

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE MÁSTER

**Diseño e implementación de un sistema de
medición y monitorización de la calidad de
servicio y experiencia de redes IPTV basadas
en el códec H.264**

Máster Universitario en Ingeniería de Telecomunicación

Alberto Mozo Robles

Tutor: López de Vergara Méndez, Jorge Enrique

Septiembre de 2016

Resumen y palabras clave

Resumen

La nueva generación de difusión y distribución de la televisión a través de la red IP (IPTV, *Internet Protocol Television*) provoca la necesidad de nuevas herramientas de control y monitorización de la calidad percibida por los usuarios de dicho servicio.

Estas herramientas de monitorización deben ser capaces de proporcionar resultados efectivos sobre la calidad de experiencia (QoE) y cumplir los requisitos de respuesta en tiempo real, de tal manera que sea posible una monitorización *online* del servicio ofrecido al cliente.

Este Trabajo Fin de Máster tiene como objetivo la implementación de una herramienta de monitorización de vídeo en alta definición (1920x1080) codificado en H.264, capaz de estimar la calidad percibida mediante la medida exclusiva de parámetros de la red (calidad de servicio, tasa de pérdidas de paquetes), siendo capaz de funcionar en tiempo real.

Se ha desarrollado una herramienta basada en un método indirecto de estimación de calidad. La herramienta precisa de información de calidad de servicio, QoS, para relacionarla con la calidad de experiencia. Para ello, es necesario la generación de un modelado matemático, mediante regresión multivariante, obteniendo un modelo de correlación entre la tasa de pérdidas y la calidad percibida. Para la generación del modelado matemático es necesario disponer de una base de datos con un número significativo de secuencias de vídeo diferentes.

Se han desarrollado dos variantes de la herramienta, en función del tipo de encapsulación empleado en la transmisión del vídeo H.264 por la red IP: RTP y MPEG-TS.

Los resultados obtenidos en la herramienta cumplen con los objetivos propuestos, consiguiendo una estimación de MOS adecuada en casi todos los casos, salvo en ciertas situaciones de alto movimiento, para ambas aproximaciones de IPTV (MPEG-TS y RTP).

Palabras clave

IPTV (Televisión IP), Calidad de Experiencia, Calidad de Servicio, monitorización, H.264, modelado matemático, regresión multivariante, VQM (*Video Quality Metric*), pérdida de paquetes.

Abstract and Keywords

Abstract

The new generation of broadcasting and distribution of television through IP networks (IPTV, Internet Protocol Television) creates the need for new tools to control and monitor the quality perceived by users of such service.

These monitoring tools must be able to provide effective results on the Quality of Experience (QoE) and meet the requirements of real-time response, so that an online monitoring service can be offered to the customer.

This master thesis aims at implementing an HD video (1920x1080) monitoring tool encoded in H.264, capable of estimating the perceived video quality by only measuring network parameters (Quality of Service, packet loss rate), being able to operate in real time.

A tool has been developed based on an indirect method to estimate the perceived quality. The tool requires information about the service quality, QoS, in order to link it with the quality of experience. For this purpose, it is necessary to develop a mathematical model, using multivariate regression model to obtain a correlation between the packet loss rate and the perceived quality. For the development of this mathematical model it is necessary to have a database with a significant number of different video sequences.

Two variants of the tool have been developed, depending on the type of encapsulation used in H.264 video transmission over the IP network: RTP and MPEG-TS.

Results obtained in the tool meet the objectives, achieving a suitable MOS estimation in almost all cases, except in the case of high degree of motion, for both IPTV approaches (Native RTP and MPEG-TS).

Keywords

IPTV (*Internet Protocol Television*), Quality of Experience, Quality of Service, monitoring, H.264, mathematics modeling, multivariate regression, VQM (*Video Quality Metric*), packet loss rate.

Agradecimientos

En primer lugar, me gustaría acordarme de todas aquellas personas que me han acompañado, ya no sólo durante la realización de este trabajo, sino de esta etapa universitaria a la que pongo fin.

Como es obvio, dar las gracias a mi tutor, Jorge Vergara, por la oportunidad ofrecida de realizar este trabajo, junto con su ayuda, dedicación y preocupación durante estos meses.

A pesar de ser un año diferente, sin clases y sin tener que acudir tanto a la Universidad, siempre es de agradecer la compañía, la ayuda incondicional y los buenos momentos junto a mis, ya ex, compañeros de clase.

También es de agradecer a todas aquellas personas que se han preocupado en saber “cómo va ese TFM”, en insistir “a ver si lo entregamos ya, ¿no?”, y en apoyarme y darme ánimos para terminar de la mejor manera posible, haciéndome sentir capaz de todo.

Por supuesto, agradecer a mi familia todas las posibilidades que me han ofrecido desde niño, con sus cuidados y preocupaciones, sus felicitaciones, ánimos y apoyo en todo momento. Son ellos los que han hecho posible que hoy esté escribiendo estas líneas.

Y no puedo olvidarme de una persona muy especial, una persona que desde hace ya unos meses está presente en todo momento a mi lado, es la persona que me ayuda cada día, con su atención infinita, su apoyo incondicional, su cariño y comprensión, hacen que me sienta con fuerzas para todo y, aunque no lo crea, tiene gran culpa de lo que aquí está escrito.

Muchas gracias.

*Alberto Mozo Robles.
Septiembre 2016*

Índice general

| | |
|--|-------------|
| ÍNDICE DE FIGURAS..... | VIII |
| ÍNDICE DE TABLAS..... | X |
| GLOSARIO | XI |
| 1. INTRODUCCIÓN | 1 |
| 1.1 MOTIVACIÓN..... | 1 |
| 1.2 OBJETIVOS | 1 |
| 1.3 PLAN DE TRABAJO..... | 2 |
| 1.4 ESTRUCTURA DEL DOCUMENTO | 3 |
| 2. ESTADO DEL ARTE..... | 5 |
| 2.1 INTRODUCCIÓN | 5 |
| 2.2 IPTV (<i>INTERNET PROTOCOL TELEVISION</i>) | 5 |
| 2.2.1 <i>H.264/AVC</i> | 6 |
| 2.2.2 <i>MPEG-TS</i> | 9 |
| 2.2.3 <i>Encapsulación IPTV H.264</i> | 13 |
| 2.2.3.1 <i>MPEG-TS sobre IP</i> | 14 |
| 2.2.3.2 <i>RTP Nativo</i> | 14 |
| 2.3 CALIDAD PERCIBIDA | 17 |
| 2.3.1 <i>Medida de la Calidad Percibida en Servicios de Telecomunicación</i> | 17 |
| 2.3.1.1 <i>VQM</i> | 20 |
| 2.4 CONCLUSIONES..... | 21 |
| 3. ANÁLISIS DEL PROBLEMA | 23 |
| 3.1 INTRODUCCIÓN | 23 |
| 3.2 ANÁLISIS DE REQUISITOS | 23 |
| 3.2.1 <i>Requisitos funcionales</i> | 23 |
| 3.2.2 <i>Requisitos no funcionales</i> | 25 |
| 3.3 CONCLUSIONES..... | 26 |
| 4. DESARROLLO DE LA SOLUCIÓN | 27 |
| 4.1 INTRODUCCIÓN | 27 |
| 4.2 GENERACIÓN DE LAS MUESTRAS DE ENTRENAMIENTO | 27 |
| 4.3 DESARROLLO DEL MODELO DE CORRELACIÓN | 30 |
| 4.3.1 <i>Ajuste sobre PLR</i> | 30 |
| 4.3.2 <i>Ajuste sobre IPB</i> | 32 |
| 4.4 DECODIFICACIÓN Y EXTRACCIÓN DE CARACTERÍSTICAS..... | 40 |
| 4.5 CÁLCULO DE LA MEDIDA DE CALIDAD DE EXPERIENCIA (QoE)..... | 42 |
| 4.6 CONCLUSIONES..... | 43 |
| 5. VALIDACIÓN Y RESULTADOS | 45 |
| 5.1 INTRODUCCIÓN | 45 |
| 5.2 EVALUACIÓN DEL MODELO DE CORRELACIÓN..... | 45 |
| 5.2.1 <i>Ajuste sobre PLR</i> | 46 |
| 5.2.2 <i>Ajuste sobre IPB</i> | 50 |
| 5.3 EVALUACIÓN DEL DECODIFICADOR DE PAQUETES IP | 54 |
| 5.4 EVALUACIÓN GLOBAL DEL SISTEMA | 56 |
| 5.4.1 <i>Secuencia con poco movimiento</i> | 56 |
| 5.4.2 <i>Secuencia con mucho movimiento</i> | 58 |
| 5.5 CONCLUSIONES..... | 59 |

| | |
|---|--------------|
| 6. CONCLUSIONES Y TRABAJOS FUTUROS | 61 |
| 6.1 RESUMEN DEL TRABAJO | 61 |
| 6.2 CONCLUSIONES..... | 61 |
| 6.3 TRABAJOS FUTUROS | 62 |
| REFERENCIAS | 63 |
| ANEXOS | LXVI |
| A.1 BITSTREAM | LXVI |
| A.1.1 RTP | LXVI |
| <i>A.1.1.1 NAL Unit Type</i> | <i>LXVI</i> |
| A.1.2 MPEG-TS | LXVII |
| <i>A.1.2.1 Stream ID</i> | <i>LXVII</i> |

Índice de figuras

| | |
|--|----|
| FIGURA 1.1. CRONOGRAMA DEL TRABAJO FINAL DE MÁSTER | 2 |
| FIGURA 2.1 SERVICIO IPTV | 6 |
| FIGURA 2.2: ESTRUCTURA CABECERA NAL UNIT | 6 |
| FIGURA 2.3: COMPRESIÓN IPB [18]..... | 7 |
| FIGURA 2.4: GRUPO DE IMÁGENES (GOP) | 8 |
| FIGURA 2.5: PERFILES H.264 [21] | 9 |
| FIGURA 2.6: ENCAPSULACIÓN DE TRANSPORT STREAM [10] | 10 |
| FIGURA 2.7: FORMATO DE UN PAQUETE MPEG PES [10] | 10 |
| FIGURA 2.8: FORMATO DETALLADO DE UN PAQUETE MPEG PES [10]..... | 11 |
| FIGURA 2.9: FORMATO DE UN PAQUETE MPEG TRANSPORT STREAM [10]..... | 11 |
| FIGURA 2.10: FORMATO DETALLADO DE UN PAQUETE MPEG TRANSPORT STREAM [10] | 12 |
| FIGURA 2.11: FORMACIÓN TS A PARTIR DE PES (MPEG-TS) [10] | 12 |
| FIGURA 2.12: EJEMPLO ASOCIACIÓN PAT-PMT-ES | 13 |
| FIGURA 2.13: PAQUETE IP + UDP + RTP | 13 |
| FIGURA 2.14: ESTRUCTURA CABECERA RTP..... | 14 |
| FIGURA 2.15: FRAGMENTACIÓN UNIDAD NAL DE UNA SLICE H.264..... | 15 |
| FIGURA 2.16: ESTRUCTURA FU (UNIDAD NAL FRAGMENTADA) | 15 |
| FIGURA 2.17. FORMATO DEL FU HEADER..... | 16 |
| FIGURA 2.18. H.264 CON MPEG-TS FRENTE A H.264 CON RTP NATIVO [2] | 16 |
| FIGURA 2.19: PROCESO DE MEDICIÓN CALIDAD DE EXPERIENCIA EN IPTV CON MÉTODO INDIRECTO . | 20 |
| FIGURA 2.20: PROCESO DE CÓMPUTO VQM | 20 |
| FIGURA 2.21: VQM FRENTE A CALIDAD SUBJETIVA [11] | 21 |
| FIGURA 3.1. DIAGRAMA DE CONTEXTO DESARROLLO DE LA HERRAMIENTA | 23 |
| FIGURA 4.1. DIAGRAMA GENERACIÓN MUESTRAS DE ENTRENAMIENTO | 27 |
| FIGURA 4.2. MODELO GILBERT-ELLIOT | 28 |
| FIGURA 4.3. MODELO SIMPLE DE GILBERT | 28 |
| FIGURA 4.4. ALGORITMO <i>DEVELOPERS MODEL</i> VQM [11] | 29 |
| FIGURA 4.5. DETALLE DEL DIAGRAMA GENERACIÓN DE LAS MUESTRAS DE ENTRENAMIENTO | 29 |
| FIGURA 4.6. DISPERSIÓN VQM MUESTRAS ENTRENAMIENTO (PLR SIMPLE) | 30 |
| FIGURA 4.7. AJUSTE MUESTRAS ENTRENAMIENTO PLR SIMPLE RTP | 31 |
| FIGURA 4.8. AJUSTE MUESTRAS ENTRENAMIENTO PLR SIMPLE MPEG-TS | 31 |
| FIGURA 4.9. DISPERSIÓN MUESTRAS ENTRENAMIENTO IL, RTP | 32 |
| FIGURA 4.10. DISPERSIÓN MUESTRAS ENTRENAMIENTO PL, RTP | 32 |
| FIGURA 4.11. DISPERSIÓN MUESTRAS ENTRENAMIENTO BL, RTP | 33 |
| FIGURA 4.12. DISPERSIÓN MUESTRAS ENTRENAMIENTO IL, MPEG-TS..... | 33 |

| | |
|---|----|
| FIGURA 4.13. DISPERSIÓN MUESTRAS ENTRENAMIENTO PL, MPEG-TS..... | 34 |
| FIGURA 4.14. DISPERSIÓN MUESTRAS ENTRENAMIENTO BL, MPEG-TS..... | 34 |
| FIGURA 4.15 AJUSTE MUESTRAS ENTRENAMIENTO EN FUNCIÓN IL, RTP..... | 35 |
| FIGURA 4.16. AJUSTE MUESTRAS ENTRENAMIENTO EN FUNCIÓN PL, RTP..... | 36 |
| FIGURA 4.17. AJUSTE MUESTRAS ENTRENAMIENTO EN FUNCIÓN BL, RTP..... | 36 |
| FIGURA 4.18. AJUSTE MUESTRAS ENTRENAMIENTO EN FUNCIÓN IL, MPEG-TS | 37 |
| FIGURA 4.19. AJUSTE MUESTRAS ENTRENAMIENTO EN FUNCIÓN PL, MPEG-TS | 37 |
| FIGURA 4.20. AJUSTE MUESTRAS ENTRENAMIENTO EN FUNCIÓN BL, MPEG-TS | 38 |
| FIGURA 4.21. AJUSTE MUESTRAS ENTRENAMIENTO MULTIVARIANTE RTP..... | 39 |
| FIGURA 4.22. AJUSTE MUESTRAS ENTRENAMIENTO MULTIVARIANTE MPEG-TS | 39 |
| FIGURA 4.23. DIAGRAMA DE IMPLEMENTACIÓN DEL DECODIFICADOR CAPTURAS IPTV | 40 |
| FIGURA 4.24. FORMATO DE SALIDA HERRAMIENTA MONITORIZACIÓN | 42 |
| FIGURA 5.1 MODELO PLR SIMPLE (ENTRENAMIENTO), RTP | 46 |
| FIGURA 5.2. MODELO PLR SIMPLE (VALIDACIÓN), RTP | 47 |
| FIGURA 5.3 MODELO PLR SIMPLE (ENTRENAMIENTO), MPEG-TS..... | 48 |
| FIGURA 5.4. MODELO PLR SIMPLE (VALIDACIÓN), MPEG-TS | 49 |
| FIGURA 5.5 MODELO PLR SIMPLE (ENTRENAMIENTO), RTP | 50 |
| FIGURA 5.6. MODELO IPB SIMPLE (VALIDACIÓN), RTP | 51 |
| FIGURA 5.7 MODELO IPB (ENTRENAMIENTO), MPEG-TS..... | 52 |
| FIGURA 5.8. MODELO IPB (VALIDACIÓN), MPEG-TS | 53 |
| FIGURA 5.9. HISTOGRAMA ERROR TASA PLR | 55 |
| FIGURA 5.10 HISTOGRAMA ERROR TASA IL | 55 |
| FIGURA 5.11 HISTOGRAMA ERROR TASA PL | 55 |
| FIGURA 5.12. HISTOGRAMA ERROR TASA BL | 56 |
| FIGURA 5.13. SECUENCIA CON POCO MOVIMIENTO, INTERVALO 113-211..... | 56 |
| FIGURA 5.14. SECUENCIA CON POCO MOVIMIENTO, FRAME 209 | 56 |
| FIGURA 5.15. SECUENCIA CON POCO MOVIMIENTO, INTERVALO 501-606..... | 57 |
| FIGURA 5.16. SECUENCIA CON POCO MOVIMIENTO, FRAME 557 | 57 |
| FIGURA 5.17. SECUENCIA CON POCO MOVIMIENTO, INTERVALO 690-788..... | 57 |
| FIGURA 5.18. SECUENCIA POCO MOVIMIENTO, FRAME 780 | 57 |
| FIGURA 5.19. SECUENCIA CON MUCHO MOVIMIENTO, INTERVALO 1-251..... | 58 |
| FIGURA 5.20. SECUENCIA CON MUCHO MOVIMIENTO, FRAME 69..... | 58 |
| FIGURA 5.21. SECUENCIA CON MUCHO MOVIMIENTO, INTERVALO 251-501 | 58 |
| FIGURA 5.22 SECUENCIA CON MUCHO MOVIMIENTO, FRAME 371..... | 59 |

Índice de tablas

| | |
|--|----|
| TABLA 2.1. ESCALA MOS (<i>MEAN OPINION SCORE</i>) | 18 |
| TABLA 4.1. BASE DE DATOS VÍDEO HDTV | 30 |
| TABLA 4.2. ESTADÍSTICAS DE CALIDAD DE SERVICIO: PÉRDIDAS | 42 |
| TABLA 4.3. ESTADÍSTICAS DE CALIDAD DE SERVICIO (QOS) FRENTE A ESTADÍSTICAS DE CALIDAD DE EXPERIENCIA (QOE) ESTIMADAS | 43 |
| TABLA 5.1. MUESTRAS DE ENTRENAMIENTO RTP (PLR SIMPLE)..... | 46 |
| TABLA 5.2. MUESTRAS VALIDACIÓN RTP (PLR SIMPLE) | 47 |
| TABLA 5.3. MUESTRAS ENTRENAMIENTO MPEG-TS (PLR SIMPLE) | 48 |
| TABLA 5.4. MUESTRAS VALIDACIÓN MPEG-TS (PLR SIMPLE)..... | 48 |
| TABLA 5.5 COMPARATIVA RTP VS MPEG-TS (PLR SIMPLE) | 49 |
| TABLA 5.6. MUESTRAS ENTRENAMIENTO RTP (IPB)..... | 50 |
| TABLA 5.7. MUESTRAS VALIDACIÓN RTP (IPB) | 51 |
| TABLA 5.8. MUESTRAS ENTRENAMIENTO MPEG-TS (IPB) | 52 |
| TABLA 5.9. MUESTRAS VALIDACIÓN MPEG-TS (IPB)..... | 52 |
| TABLA 5.10 COMPARATIVA RTP VS MPEG-TS (IPB) | 54 |
| TABLA 5.11. COMPARATIVA PLR VS IPB | 54 |
| TABLA 5.12 ERROR MEDICIÓN TASAS DE PÉRDIDAS..... | 55 |

Glosario

TIC: Tecnologías de la Información y la Comunicación.

IPTV: *Internet Protocol Television*, difusión de Televisión a través de las redes de banda ancha IP.

QoS: *Quality of Service*, Calidad de Servicio, rendimiento de una red. Cuantitativamente mide la calidad de los servicios de red, como son la tasa de pérdidas, ancho de banda, retraso, disponibilidad, jitter, etc.

QoE: *Quality of Experience*, Calidad de Experiencia, nivel o grado de satisfacción de un usuario, medido de forma subjetivo, en el uso de un servicio multimedia.

VQM: *Video Quality Metric*, método estándar para la medición de la calidad del vídeo de forma objetiva.

MOS: *Mean Opinion Score*, medida ponderada y subjetiva que representa la calidad percibida por los usuarios.

PLR: *Packet Loss Rate*, tasa de pérdida de paquetes.

MPEG IBP: Frames I (Intra), P (Predictive) y B (Bidirectional) del estándar MPEG con el que son codificadas las imágenes de vídeo.

1. Introducción

1.1 Motivación

Las Tecnologías de la Información y la Comunicación (TIC) están presentes en todos los ámbitos en el día a día de las personas, incorporadas no sólo en el desempeño laboral, sino también en las actividades lúdicas, en el llamado “ocio digital”.

El gran incremento en el consumo de elementos tecnológicos incentiva, a su vez, la evolución y regeneración de las propias tecnologías, buscando su adaptación a los nuevos requisitos que demandan los cada vez más exigentes usuarios.

De esta manera, surge una nueva generación de redes IP (IPTV *Televisión IP*) que permite una nueva forma de difusión y distribución de la televisión en red, mediante las conexiones residenciales de banda ancha sobre el protocolo IP. Este hecho despierta interés en los proveedores de servicios de telecomunicaciones, cuyo negocio dependerá, en gran medida, de la calidad proporcionada a los clientes, por lo que surge la necesidad de disponer de herramientas capaces de llevar a cabo la monitorización de las redes multimedia.

Dichas herramientas deben implementar métodos efectivos para medir la calidad experimentada por los usuarios (QoE). Los métodos que utilizan el vídeo original para realizar las mediciones y comparativas con el vídeo recibido por los usuarios (vídeo degradado) son los que mejor estiman la calidad percibida, si bien son métodos irrealizables en monitorización en tiempo real, por el hecho de precisar del vídeo original y por su elevado coste computacional.

Por todo ello, se hace necesario disponer de alternativas capaces de estimar la calidad de vídeo percibida por los usuarios de IPTV, sin necesitar referencia, únicamente a través de medidas de calidad de la red (QoS), sirviendo así respuestas inmediatas y fiables, aplicables en tiempo real. El aspecto novedoso de este trabajo es la aplicación de las medidas de calidad de experiencia vídeo de alta definición (HDTV: 1920x1080 píxeles) con codificación H.264 sobre RTP.

1.2 Objetivos

El objetivo principal de este Trabajo Final de Máster es implementar una herramienta capaz de estimar la calidad de vídeo percibida por un usuario de IPTV mediante la medida de parámetros de la red (pérdida de paquetes, QoS), aplicada a vídeo de alta definición, codificado en H.264 (códec de vídeo para alta definición), capaz de funcionar en monitorización en tiempo real.

Para la generación de dicha herramienta, será necesario cumplir una serie de objetivos parciales, garantizando así la fiabilidad de la misma:

- Construcción y generación de un subconjunto de muestras de test y entrenamiento: tasas de pérdida de paquetes (QoS) de secuencias de vídeo originales y degradadas, generadas mediante la herramienta de software libre *Netem* [17], junto con su respectiva calidad de experiencia (QoE), mediante el algoritmo VQM [11].
- Desarrollo y construcción de un modelo matemático para el ajuste de los datos de test: calidad de servicio (tasa de pérdidas) frente a calidad de experiencia (VQM). Para la

generación del modelo, se tendrá en cuenta información característica de la codificación MPEG-4, parte 10 (H.264) [5].

- Implementación de un decodificador de paquetes IP, capaz de identificar y detectar flujos RTP que transportan información de cuadros de vídeo codificado en H.264, para la estimación y medida de tasas de pérdidas.
- Desarrollo de un módulo encargado de determinar y estimar la calidad de experiencia en escala MOS, tomando como entrada un flujo de paquetes IP en un archivo *pcap*.

1.3 Plan de trabajo

El desarrollo de este proyecto se ha estructurado en las siguientes fases:

- Fase I: Análisis de la situación.
 - Estudio y evaluación del estado actual de la tecnología IPTV.
 - Análisis profundo del estándar de codificación de alta definición HDTV: H.264, junto con el estándar MPEG-TS.
 - Análisis del estándar de transmisión de audio y vídeo: MPEG-TS.
 - Estudio de trabajos previos relacionados.
 - Análisis sobre diferentes métodos y algoritmos de estimación de calidad percibida en vídeo (tales como VQM).
- Fase II: Diseño de la solución.
 - Identificación y selección de las herramientas tecnológicas y algoritmos necesarios para la ejecución del trabajo.
 - Obtención de datos fuente: secuencias de vídeo en alta definición codificadas en H.264 con diferentes tasas de muestreo.
- Fase III: Ejecución y desarrollo.
 - Generación del subconjunto de datos de test y entrenamiento.
 - Ajuste del modelo matemático QoS-QoE.
 - Implementación del decodificador de paquetes IP.
 - Construcción del módulo de respuesta de calidad de experiencia (QoE).
- Fase IV: Validación.
 - Contraste y validación de los resultados obtenidos.
- Fase V: Documentación.
 - Redacción de la documentación y memoria del trabajo, plasmando el conocimiento y experiencia adquiridos durante la realización del Trabajo Final de Máster.

La planificación del Trabajo Final de Máster se ve plasmada en el siguiente cronograma (Figura 1.1), en la que se observa una duración y esfuerzo en 300 horas totales, siendo la tarea de Ejecución y Desarrollo la principal del mismo, con 225 horas aproximadas).

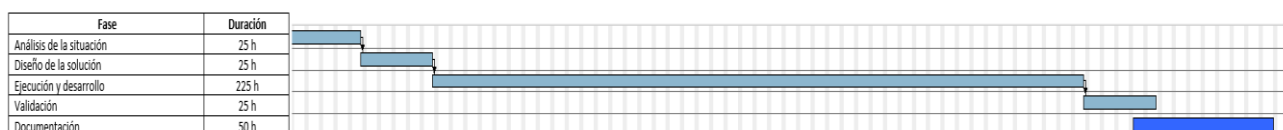


Figura 1.1. Cronograma del Trabajo Final de Máster

1.4 Estructura del documento

Este documento refleja todo el proceso de aprendizaje y desarrollo del Trabajo Final de Máster. Se encuentra organizado de la siguiente manera:

- En el presente capítulo se ha realizado una introducción al tema tratado en el proyecto, así como una motivación de por qué se considera necesario el desarrollo de una solución tecnológica al respecto.
- En el capítulo 2 se expone el estado del arte de la tecnología IPTV, del estándar de codificación para la alta definición, H.264, y para la transmisión de audio y vídeo, MPEG-TS, así como de diferentes métodos y algoritmos de estimación de calidad percibida por el usuario, entre los que destaca VQM.
- En el capítulo 3 se especifican y detallan los requisitos, tanto funcionales como no funcionales, que debe cumplir la herramienta implementada, con el fin de cumplir los objetivos propuestos.
- En el capítulo 4 se explica detalladamente el diseño de la herramienta, así como el procedimiento necesario para su implementación.
- En capítulo 5 se realiza una evaluación y validación de la herramienta implementada, mostrando los resultados obtenidos por la misma.
- Por último, en el capítulo 6 se plasman las conclusiones obtenidas, así como las propuestas de trabajo futuro y mejoras, generadas tras la realización de este trabajo.

2. Estado del arte

2.1 Introducción

Este capítulo de la memoria resulta de vital importancia en el desarrollo del Trabajo Final de Máster, puesto que nos ayuda a comprender el problema tratado y conocer las soluciones existentes, evitando así desarrollos innecesarios y errores previos.

Se explicarán las bases tecnológicas de la transmisión IPTV. Del mismo modo, se detallarán las especificaciones de los estándares de codificación y transmisión que aplican, siendo H.264 el estándar de codificación empleado para los vídeos de alta definición, y MPEG-TS el estándar de transmisión empleado en IPTV. También se revisará el concepto de “calidad percibida” en relación con los servicios de telecomunicaciones, así como los diferentes algoritmos y métodos de estimación de calidad, centrándonos en el método VQM (*Video Quality Models*).

2.2 IPTV (*Internet Protocol Television*)

Dentro del concepto de ‘Vídeo sobre IP’ se distinguen dos entornos:

1. **Servicio de televisión sobre protocolo IP (IPTV).** Se trata de servicios de televisión de pago, ofrecidos por operadores de telecomunicación, en los cuales la señal de televisión se transmite a través de las infraestructuras cableadas utilizadas para los servicios de internet (cobre o fibra). El contenido se distribuye por redes privadas, seguras y dedicadas, administradas y gestionadas por las operadoras del servicio. Es necesario una suscripción y un decodificador de la señal (STB: *Set-top-box*). A este tipo de servicio vídeo sobre IP se le conoce con el concepto de “*Walled garden*”, debido a la seguridad y exclusividad del cliente con el ISP (Proveedor de Servicios de Internet).
2. **Servicio de difusión de vídeo y televisión por Internet (*Streaming*).** Es un servicio de transmisión de televisión distribuido a través de Internet, consistente en la descarga continua del contenido multimedia a través de una red abierta, sin gestión de calidad ni necesidad de contratación. Este otro servicio de televisión por Internet se denomina una solución *Over-The-Top* (OTT), soluciones que aprovechan el poder de Internet para ofrecer servicios sofisticados, como el *streaming*.

El estudio de este Trabajo Final de Máster se centra en los Servicios de IPTV, ya que son aquellos en los que existe un mayor interés en la satisfacción del usuario y, por lo tanto, en la calidad ofrecida y percibida por los mismos.

El Servicio IPTV (figura 2.1) diferencia entre los servicios de difusión (Canales de TV) distribuidos a través de la red como *multicast*, y los servicios bajo demanda (VoD), entregados al hogar en *unicast*.

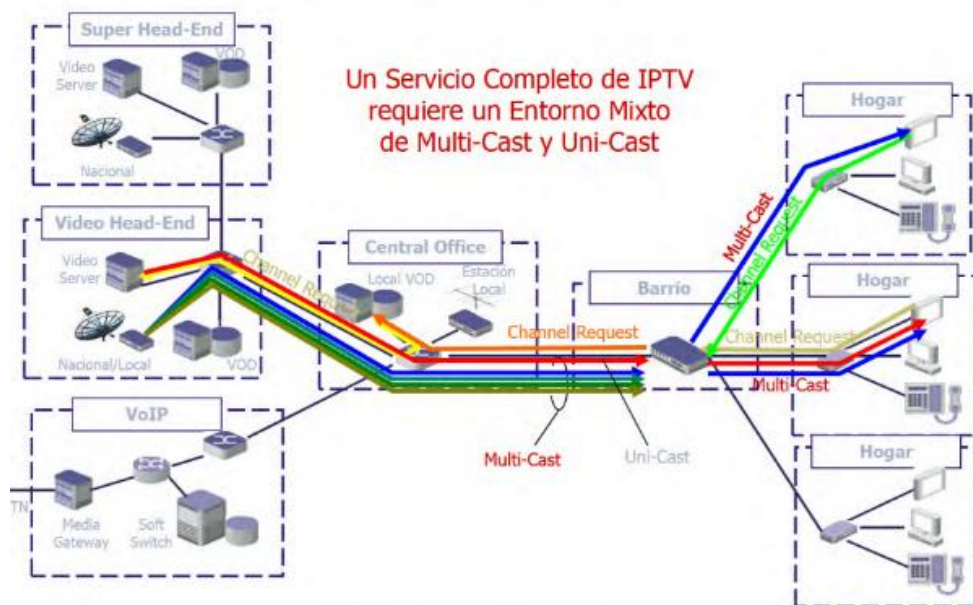


Figura 2.1 Servicio IPTV

2.2.1 H.264/AVC

H.264/MPEG-4 parte 10/MPEG-4 AVC es un estándar de codificación de vídeo de alta compresión desarrollado conjuntamente por el ITU-T Video Coding Experts Group (VCEG) y el ISO/IEC Moving Picture Experts Group (MPEG) [5][8].

El estándar está diseñado de forma genérica, siendo capaz de cubrir un amplio rango de aplicaciones (comunicaciones en tiempo real, difusión de televisión, etc.), tasas de bit, resoluciones, calidades y servicios.

H.264 es un estándar de codificación compatible con todas las técnicas más eficientes de compresión de vídeo existentes. Es capaz de reducir el tamaño de un archivo de vídeo en más de un 80%, comparado con Motion JPEG, y en más de un 50% comparado con MPEG-2. Este hecho implica que se requiere un menor ancho de banda y espacio de almacenamiento para los archivos de vídeo, permitiendo asimismo el desarrollo de la televisión en alta definición (HDTV).

La capa de abstracción de red (*Network Abstraction Layer*, NAL) es la parte del estándar cuyo objetivo es facilitar una representación del contenido a codificar. Los datos de vídeo codificado se organizan en unidades NAL (*NAL unit*). El primer byte de cada unidad NAL es un byte de cabecera (*NAL header*), que indica el tipo de datos que contiene dicha unidad NAL, la presencia de errores en los datos e información de prioridad en el proceso de decodificación (figura 2.2). El resto de bytes de la unidad NAL son la carga útil (*NAL payload*).

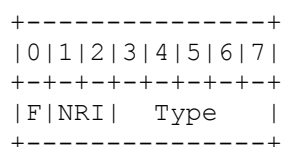


FIGURA 2.2: ESTRUCTURA CABECERA NAL UNIT

El formato de la cabecera de una unidad NAL es el mostrado en la figura 2.2. con los campos:

F: *forbidden_zero_bit*: toma el valor 1 cuando existe una violación de sintaxis.

NRI: *nal_ref_idc*: indica información sobre la referencia para la predicción de la información de los cuadros de vídeo o *frames*. 00 indica que el contenido de la unidad NAL no se usa para reconstruir otros *frames*.

Type: *nal_unit_type*: indica el tipo de unidad NAL. Se informa si la unidad NAL codifica información de una imagen IDR o imagen no IDR. Una imagen IDR es aquella que no contiene predicción de ningún tipo (imágenes tipo I o tipo SI). Los tipos de unidades NAL se indican en el Anexo A.1.1.

Las unidades NAL forman los fragmentos básicos de vídeo transmitidos.

H.264/MPEG-4 AVC especifica el mismo tipo de imágenes que en las normas anteriores (I, P, B), introduciendo dos nuevas: SI (Switching I) y SP (Switching P), empleadas para la transición entre flujos de vídeo, sin necesidad de enviar imágenes I pesadas y utilizando predicción espacial y temporal. Este tipo de imágenes no se utilizan en la difusión de televisión, por lo que no serán de especial análisis.

- Imágenes I: (*intraframed-coded*): se trata de imágenes codificadas de manera independiente, sin referencia a ninguna otra imagen, por lo que no se emplean técnicas de predicción temporal. El uso de las imágenes I de forma periódica facilita la reinicialización del receptor, además de corregir los errores producidos por pérdidas de imágenes anteriores. La periodicidad de las imágenes I puede variar en función del codificador.
- Imágenes P: (*predictive-coded*): son imágenes codificadas como las diferencias entre la imagen predicha y la de referencia, por lo que aprovecha semejanzas entre imágenes próximas temporalmente. Las predicciones sólo se realizan hacia adelante, es decir, las predicciones se forman únicamente con información de imágenes I o P anteriores en el tiempo. También es posible tomar como referencia imágenes anteriores de tipo B.
- Imágenes B: (*bidirectionally predictive-coded*): son imágenes comprimidas de manera muy similar a las P, codificando la diferencia entre una imagen predicha y una imagen de referencia I o P. La novedad de H.264 es que también es posible emplear imágenes B como referencia. La diferencia con las imágenes P es que la predicción se obtiene de imágenes cercanas en la secuencia, I o P, tanto pasadas como futuras. El número de cuadros B entre dos imágenes de referencia es un parámetro determinado por el codificador.

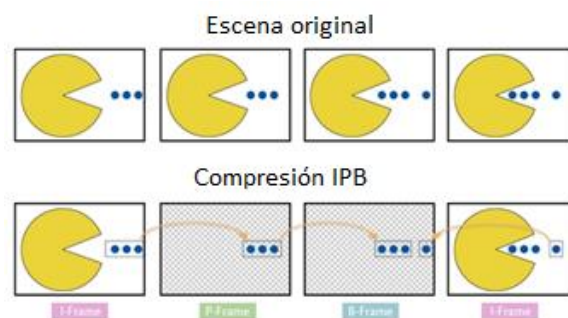


Figura 2.3: Compresión IPB [18]

De acuerdo con lo explicado anteriormente, las imágenes se organizarían tal y como se muestra en la figura 2.4. A este conjunto se lo denomina GOP (*Group of Pictures*, Grupo de Imágenes). La estructura GOP especifica el orden en el que las imágenes se organizan. De esta manera, una secuencia de vídeo codificada es una sucesión de GOPs. Debe comenzar siempre por una imagen de tipo I y termina cuando se encuentra una nueva imagen I. De esta manera, cualquier error es corregido con la decodificación de la siguiente imagen I.

Los GOPs pueden ser abiertos o cerrados. Un GOP abierto es aquel en el que las predicciones en las imágenes P y B pueden realizarse con imágenes de otros GOPs, mientras que un GOP cerrado las imágenes P y B sólo se predicen de las del mismo grupo. Este concepto se relaciona con el concepto de imagen IDR.

Una imagen IDR (*Instantaneous Decoding Refresh*) es una imagen de tipo I, a partir de la cual cualquier imagen P o B sólo puede tomar referencia de imágenes posteriores a la imagen I. Por lo tanto, un GOP cerrado es aquel en el que las imágenes I son de tipo IDR.

Las imágenes IDR también se relacionan con el término de acceso aleatorio, puesto que un reproductor podrá empezar siempre que encuentra una imagen IDR, dado que no habrá ninguna imagen que necesite información de imágenes previas al IDR. Por lo tanto, las imágenes IDR son los puntos de acceso y de inicio de decodificación.

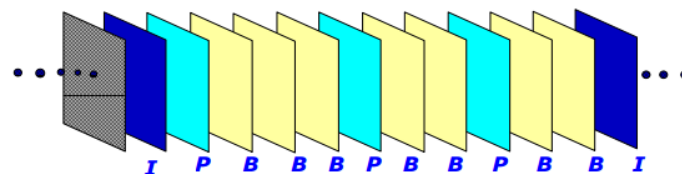


Figura 2.4: Grupo de imágenes (GOP)

En el estándar H.264/MPEG-4 AVC los codificadores tienen mucha flexibilidad para elegir la estructura de referencia del GOP. Un GOP queda determinado por dos parámetros:

- El tamaño del GOP, definido como la diferencia de imágenes entre dos imágenes I, como *Size of Coded Video Sequence*.
- El parámetro de *B-frames*, que indica el número de imágenes B entre imágenes I, P.

No obstante, el codificador tiene la capacidad de detectar cambios de secuencia o escena, en los cuales introduce imágenes de tipo I, que es la mejor imagen en esos casos, y sirve como un punto de referencia (*Adaptive B-frames placement*). De esta manera, el codificador modifica el número de imágenes B cuando se mejora la calidad en la codificación.

En el estándar H.264, la granularidad para establecer predicciones es una capa inferior a la imagen, denominada *slice* (tira). Una *slice* es una región espacial de una imagen que es codificada de forma totalmente independiente de otras regiones de la misma imagen. Por ello, los conceptos de imágenes I, P y B son trasladados a *slice I*, *slice P* y *slice B*.

En función de las necesidades y capacidades de cada transmisión, se determinan una serie de parámetros de codificación: concepto de “perfiles” y “niveles”. El uso de perfiles y niveles permite una interoperabilidad entre distintas implementaciones de decodificadores.

Cada perfil especifica un conjunto de características de algorítmica que deben cumplirse y ser soportadas por todos los decodificadores que cumplen dicho perfil. Cada perfil define el conjunto de características que puede usar el codificador y limita la complejidad de implementación del mismo.

H.264 tiene siete perfiles, cada uno de los cuales va destinado a una clase concreta de aplicaciones. Cada perfil define qué conjunto de características puede usar el codificador y limita la complejidad de implementación del decodificador.

Cada nivel especifica los límites y los valores que deben tomar los elementos de sintaxis del estándar. Los mismos niveles aplican a diferentes perfiles. Los niveles se corresponden con las capacidades de memoria y carga de procesamiento del decodificador. Por cada nivel se especifican características como tamaño de la imagen (QCIF, CIF, 480i, 576i, 720p, 1080i, 1080p, 2kx1k, 4kx2k), tasa de imágenes (*frame rate*), máxima tasa de bit (*bit rate*) y número máximo de imágenes de referencia.

Perfiles:

- Baseline profile: es el perfil de mínimo coste computacional y máxima robustez frente a errores. Se emplea en los teléfonos móviles, videotelefonía, videoconferencia y comunicaciones inalámbricas. No se permite el uso de imágenes B, eliminando así latencia, por lo que se utiliza también en entornos de videovigilancia.
- Main profile: perfil con máxima tasa de compresión y menor énfasis en robustez. Empleado en la difusión de TV y almacenamiento (Blu-ray).
- Extended profile: este perfil contiene al perfil Baseline y lo complementa con características del perfil principal (Main profile), consiguiendo un compromiso entre ambos, con atención especial a las aplicaciones de vídeo streaming.
- FRExt: extensión del perfil: alcanzar la alta fidelidad [20]

Los componentes que forman parte de cada perfil se muestran en la figura 2.5.

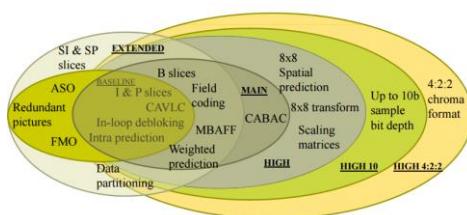


Figura 2.5: Perfiles H.264 [21]

2.2.2 MPEG-TS

MPEG Transport Stream [10] es un estándar para la transmisión y almacenamiento de audio y vídeo. Se utiliza en sistemas de difusión como DVB (*Digital Video Broadcasting*), ATSC e IPTV.

Especifica un formato contenedor, encapsulando los datos de vídeo y audio en paquetes (PES, *Packetized Elementary Stream*). Los flujos binarios de vídeo y audio de cada canal de televisión se comprimen independientemente, formando los llamados ES (*Elementary Stream*), estructurándose a su vez en forma de paquetes (PES), tal y como se muestra en la figura 2.6. En el caso de una secuencia de vídeo, cada imagen se corresponde con un ES.

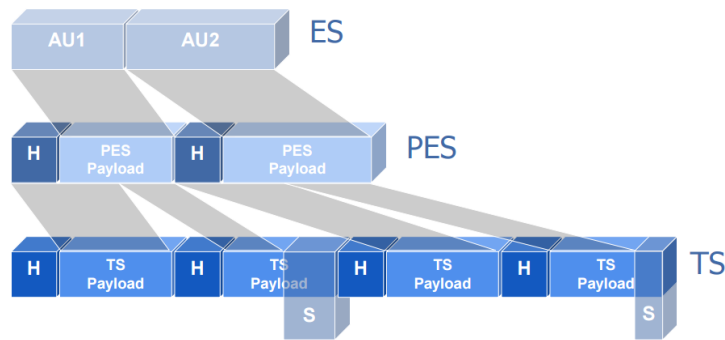


Figura 2.6: Encapsulación de Transport Stream [10]

Los paquetes de audio y vídeo (PES), junto con otros datos de señalización y sincronización del protocolo, se multiplexan, formándose un único bitstream, llamado *Transport Stream* (TS). Se utilizan paquetes de relleno (*Stuffing*) para completar el payload del TS. Cada ES conforma un programa de televisión.

Los paquetes PES, de longitud variable en función del ES que empaquetan, tienen la estructura mostrada en las figuras 2.7 y 2.8.

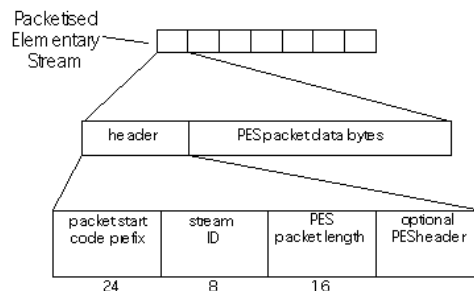


Figura 2.7: Formato de un paquete MPEG PES [10]

Cabecera de un paquete PES:

- Código prefijo de inicio de paquete: *Packet start code prefix*: 3 bytes cuyo valor es siempre 0x000001
- Identificador de stream: *Stream id*: identificador de 1 byte que describe el tipo de datos que contiene. Están especificados en el Anexo A.1.1.
- Longitud de paquete: *PES packet length*: 2 bytes que indican la longitud del paquete PES. El valor puede ser 0 sólo cuando el payload del paquete PES es un ES de vídeo.
- Campos opcionales de la cabecera, entre los que destacan:
 - *PES scrambling control*: modo de cifrado de la carga útil del PES.
 - *Data alignment indicator*: si el valor es 1, indica que inmediatamente después de la cabecera se encuentra el código de sincronización de audio o vídeo.
 - *Copyright*: si el valor es 1, el material incluido en la carga útil está protegido por copyright.
 - *PTS and DTS (Presentation and Decoding TimeStamp)*: tiempo de presentación y decodificación del paquete PES.
 - *ES rate*: representa la tasa a la que el decodificador recibe la información.

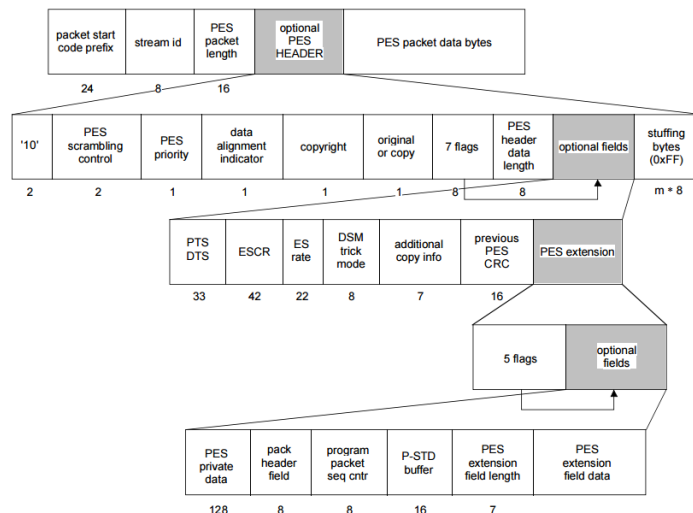


Figura 2.8: Formato detallado de un paquete MPEG PES [10]

Los paquetes TS son de longitud constante, 188 bytes, de los cuales 184 bytes son de carga útil (*payload*) y 4 bytes de cabecera (figuras 2.9 y 2.10).

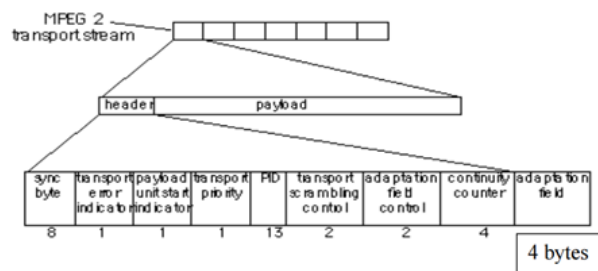


Figura 2.9: Formato de un paquete MPEG Transport Stream [10]

Cabecera de un paquete TS:

- 1) Byte de sincronización: byte que sirve para que el decodificador se sincronice correctamente. Tiene el valor 0x47 (0100 0001).
- 2) Flags: tres indicadores/flags usadas para indicar cómo debe ser procesado el payload:
 - Indicador de error de transporte: se pone a 1 cuando se detecta un error en la transmisión.
 - Indicador de arranque: indica el inicio de un paquete PES.
 - Indicador de prioridad.
- 3) Identificador del paquete, PID (*Packet Identifier*): es un campo de 13 bits que permite la identificación y distinción de paquetes de diferentes Elementary Streams. Son posibles 2^{13} valores, de los cuales 17 están reservados para funciones especiales de señalización y control del protocolo (PSI, *Program Specific Information*). Paquetes con PID desconocido o con PIDs no esperados por el decodificador son descartados.
- 4) Control de cifrado: Los dos bits de *scrambling control* se emplean en procedimientos de acceso condicional para cifrar la carga útil de los paquetes TS.
- 5) Control campo de adaptación, que puede tomar cuatro valores:
 - 01 – no hay campo de adaptación, sólo carga útil.

- 10 – sólo campo de adaptación, sin carga útil.
 - 11 – campo de adaptación + carga útil.
 - 00 – RESERVADO, para uso futuro.
- 6) Contador de continuidad, CC (*Continuity Counter*): contador incrementado por el codificador que sirve para controlar y deducir las pérdidas de paquetes TS.
- 7) Campo de adaptación, que en caso de existir, contiene los siguientes subcampos:
- Longitud del campo de adaptación
 - Indicador de discontinuidad
 - Indicador de acceso aleatorio
 - Flags
 - Campos opcionales: PCR (*Program Clock Reference*), Extensión del campo de adaptación, etc.

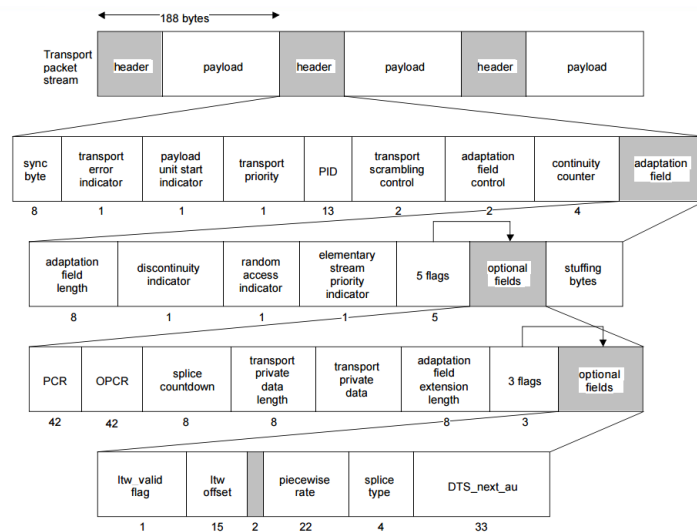


Figura 2.10: Formato detallado de un paquete MPEG Transport Stream [10]

La formación del TS a partir de paquetes PES se muestra en la figura 2.11. En ella, se aprecia la única regla de corte o unión del estándar MPEG-TS (ISO/IEC 13818-1): el primer byte de cada paquete PES se coloca en el primer byte de carga útil disponible del Transport Stream.

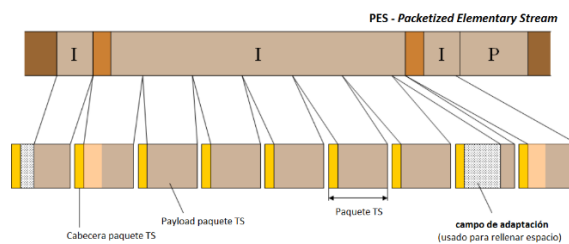


Figura 2.11: Formación TS a partir de PES (MPEG-TS) [10]

El protocolo MPEG-TS también define unas tablas de información de servicio con toda la señalización necesaria para la correcta decodificación de los flujos de audio y vídeo, PSI (*Program Specific Information*). Toda la señalización queda conformada en otros ES y se introducen en la propia trama de transporte junto con el audio y el vídeo. El estándar define una serie de tablas obligatorias:

- **PAT (Program Association Table):** contiene la información de los programas presentes en un Transport Stream. Asocia los números de programa con los PID de las tablas PMT de cada programa. La PAT tiene el PID=0x0, y es única por TS.
- **PMT (Program Map Table):** existe una tabla PMT por cada programa existente en el TS. En ella se encuentra la información de todos los ES asociados al programa. Por lo tanto, para cada ES indica:
 - PID de los ES.
 - Tipo de dato (audio, vídeo, datos, etc.)
 - Descriptores asociados.
- **CAT (Conditional Access Table):** contiene información sobre el sistema de acceso condicional presente en el Transport Stream. Por ello, sólo es obligatoria en los casos en los que exista cifrado de algún ES. La tabla CAT tiene asignado el PID=0x1.
- **NIT (Network Information Table):** transporta información de red. Es una tabla opcional, pero en el caso de estar presente siempre lleva asignado el programa número 0 de la PAT.

En la figura 2.12 vemos un ejemplo de las tablas de señalización MPEG-TS

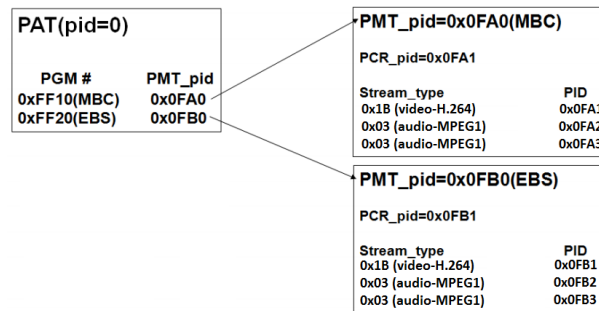


Figura 2.12: Ejemplo asociación PAT-PMT-ES

2.2.3 Encapsulación IPTV H.264

En IPTV, al tratarse de un servicio de transmisión de datos multimedia (audio + vídeo), los paquetes IP transmitidos encapsulan datagramas UDP (figura 2.13). Esto es debido a que el protocolo UDP trabaja en comunicaciones no orientadas a conexión, sin ningún tipo de sincronización entre origen y destino, permitiendo así consultas y transferencias de datos rápidas y sin retransmisiones. No obstante, este hecho implica que se trata de comunicaciones no fiables, en las que no se garantiza la entrega de los paquetes ni el orden del mismo, lo que provoca **pérdidas de paquetes** en las comunicaciones UDP.

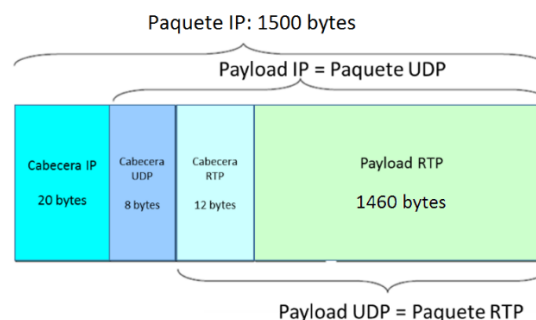


Figura 2.13: Paquete IP + UDP + RTP

Los paquetes UDP, a su vez, encapsulan mensajes RTP, protocolo para transporte de datos en tiempo real (*Real Time Transport Protocol*). RTP proporciona información de tiempo y número de secuencia, que sirve de utilidad para la sincronización y protección frente a pérdidas o desordenación de paquetes.

RTP utiliza conjuntamente un protocolo de control: RTCP (*Real-Time Transport Control Protocol*). Este protocolo proporciona mecanismos de monitorización de la calidad de servicio.

Actualmente, existen varias alternativas para transportar vídeo, ya sea H.264 o MPEG-2, sobre IP: utilizando el protocolo de transporte MPEG-TS o directamente sobre RTP.

2.2.3.1 MPEG-TS sobre IP

Utilizando el protocolo MPEG-TS tenemos dos opciones de transportar la información multimedia (audio + vídeo):

- Directamente sobre UDP: Ethernet tiene una MTU (*Maximum Transmission Unit*, Unidad Máxima de Transferencia) de 1500 bytes, y un paquete TS son 188 bytes, por lo que la carga útil de UDP se corresponde con $1500-20(IP)-8(UDP)/188 \approx 7$ paquetes TS.
- Empleando el protocolo RTP (especificado por IETF y por el grupo DVB-IP), en el que también habría 7 paquetes TS en la carga útil de RTP. En este caso, para identificar el contenido del payload RTP, se establece en la cabecera RTP el tipo de payload (*Payload type*) con valor 33 (*MPEG-II transport streams*).

2.2.3.2 RTP Nativo

El transporte de datos de vídeo codificados en H.264 es posible transmitirla mediante el protocolo RTP.

El formato de la cabecera RTP está especificado en la norma RFC 3550 y sigue el formato mostrado en la figura 2.14, destacando los campos:

M (*marker bit*): 1 bit que representa el final de una unidad NAL. Se pone a 1 cuando el paquete RTP contiene el último fragmento de la unidad NAL.

PT (*payload type*): 7 bits. Indica el tipo de payload contenido en el paquete RTP. Para el caso de H.264, el valor es 96.

SN (*sequence number*): 16 bits. Número de secuencia utilizado para determinar el orden de decodificación de las unidades NAL.

Timestamp: 32 bits. Toma el valor de tiempo de la muestra. En el caso de H.264, todas las unidades NAL correspondientes a la misma *slice* presentan la misma marca de tiempos.

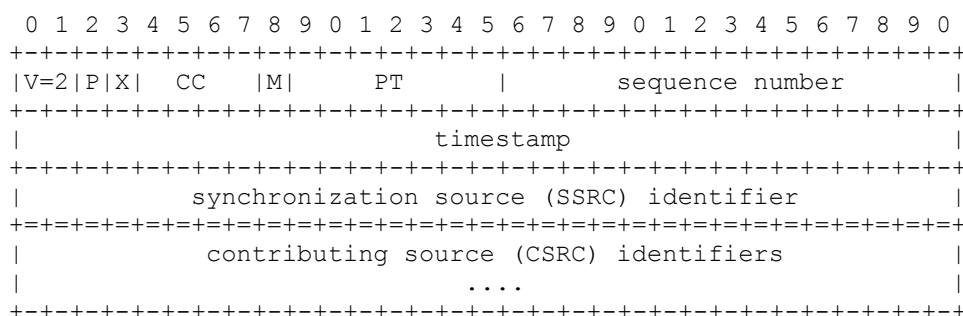


Figura 2.14: Estructura cabecera RTP

Existen tres estructuras de payload RTP conteniendo H.264. Está identificado en el primer byte del de la carga útil, por lo que sirve como una extensión a la cabecera RTP. Las tres posibles estructuras son:

- *Single NAL Unit Packet*: el paquete RTP contiene una única unidad NAL.
- *Aggregation Packet*: el payload RTP contiene agregadas múltiples unidades NAL.
- *Fragmentation Unit*: una única unidad NAL se fragmenta en múltiples paquetes RTP. FU-A y FU-B.

Se ha observado que, para los vídeos H.264 utilizados en este trabajo, la estructura que tienen los paquetes RTP es *Fragmentation Unit*. Se debe a que es necesario transportar NALU de tamaño superior a 64 kbytes, como el caso del vídeo de alta definición (figura 2.15)

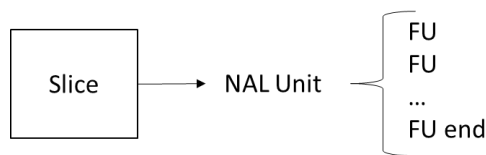


Figura 2.15: Fragmentación Unidad NAL de una slice H.264

La fragmentación se define para una única Unidad NAL. Por lo tanto, un fragmento de Unidad NAL (FU, *Fragmentation Unit*) consiste en un número entero de bytes consecutivos de la Unidad NAL. Los fragmentos de la misma Unidad NAL se transmiten en orden, con número de secuencia RTP ascendente. La marca de tiempos de cada paquete RTP que contiene FU es el mismo, correspondiente al *timestamp* de la Unidad NAL fragmentada. En la figura 2.16 se muestra el formato de una Unidad Fragmentada (FU).

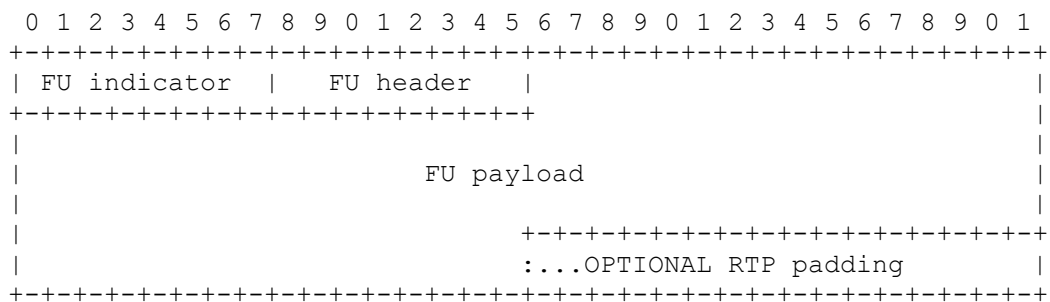


Figura 2.16: Estructura FU (Unidad NAL Fragmentada)

Una Unidad NAL Fragmentada (FU) está compuesta por:

FU indicator: un byte que indica la cabecera de la Unidad NAL (apartado 2.2.1)

FU header: byte con el formato mostrado en la figura 2.17.

S: 1 bit. Indica el inicio de la Unidad NAL fragmentada, cuando el valor es 1.

E: 1 bit. Indica el fin de la Unidad NAL fragmentada, cuando el valor es 1.

R: 1 bit. Es el bit de reserva. Su valor debe ser 0 (es ignorado por el receptor).

Type: *nal_unit_type*. Tipo de Unidad NAL (Anexo A.1.1)

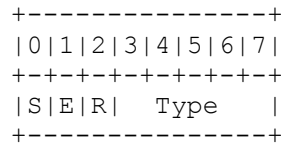


Figura 2.17. Formato del FU header

En la figura 2.18 se muestra una comparativa entre el uso de MPEG-TS frente al uso de RTP Nativo para la transmisión de vídeo H.264. La razón más importante para evitar la transmisión TS/UDP/IP o TS/RTP/UDP/IP, es decir, MPEG-TS, es la susceptibilidad de este mecanismo de transporte a los errores resultantes de la pérdida de paquetes.

H.264 proporciona varias técnicas para ofrecer robustez frente a pérdidas, ordenando y codificando las Unidades NAL de forma inteligente, entrelazando los paquetes, consiguiendo así una mayor robustez en el caso de pérdidas consecutivas de paquetes. Por otro lado, MPEG-TS necesita una codificación secuencial de los PES en un paquete TS.

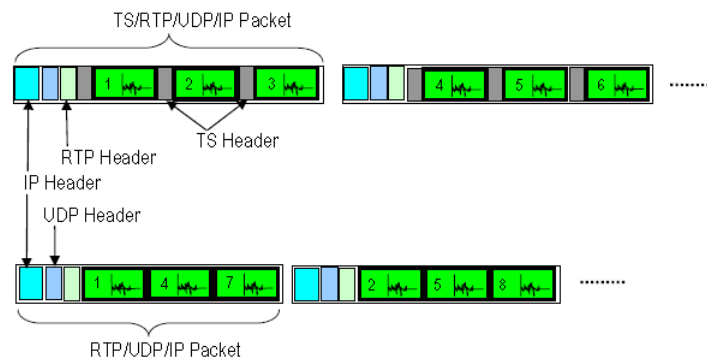


Figura 2.18. H.264 con MPEG-TS frente a H.264 con RTP Nativo [2]

Por este motivo, en el caso de producirse una pérdida de un datagrama UDP, en el ejemplo mostrado en la figura 2.18, perderíamos los paquetes 1, 2 y 3 con MPEG-TS, es decir, tres paquetes consecutivos, caso que tiene mayor repercusión visual que si perdemos paquetes no consecutivos (1, 4 y 7 para el caso de RTP Nativo). Este hecho explicará que en los modelados de pérdidas de paquete frente a calidad de experiencia (apartado 4.3) tengamos casos con pérdidas altas y calidad aceptable en el caso de RTP Nativo que no están presentes en el caso de MPEG-TS.

Por otro lado, existen diferentes aplicaciones en las que los clientes requieren diferentes combinaciones para el mismo contenido: seleccionar el lenguaje del audio, los subtítulos, diferentes vistas de cámara, etc. En estos casos, emplear RTP Nativo permite servir de forma independiente a cada cliente el flujo requerido. Con MPEG-TS la alternativa sería enviar todas los ES y que sea el cliente quien seleccione los que le interese, provocando un consumo excesivo de ancho de banda.

Otro servicio en el que se hace necesario emplear RTP sin MPEG-TS es el *trick-play*, servicio de vídeo bajo demanda (VoD, *Video on Demand*) en el que el cliente tiene la capacidad de pausar, reanudar, acelerar y rebobinar la reproducción del vídeo.

2.3 Calidad percibida

Una vez visto y revisado los estándares de codificación de vídeo de alta definición, pasamos a estudiar cómo se modela la calidad de vídeo con la que los clientes de IPTV visualizan el vídeo.

Se entiende como “calidad percibida” a la Calidad de Servicio Percibida por los usuarios del servicio multimedia, denominada “Calidad de Experiencia” (QoE). Se trata de una calidad subjetiva que no es posible medir únicamente con parámetros de rendimiento de la red de paquetes.

La ITU-T G.100/P.10 [19] define la Calidad de Experiencia como “la aceptabilidad general de una aplicación o servicio, tal y como la percibe subjetivamente los usuarios finales”, aclarando que la Calidad de Experiencia incluye todos los efectos del sistema extremo a extremo (cliente, equipo terminal, red, infraestructura de servicios, etc.), y que la aceptabilidad general puede estar influenciada por el contexto y las expectativas del usuario [3].

La ETSI TISPAN (*European Telecommunications Standards Institute Telecommunications and Internet converged Services and Protocols for Advanced Networking*) la define como “la experiencia percibida por el usuario que está haciendo uso de servicios de comunicaciones”.

Por lo tanto, la Calidad Percibida o Calidad de Experiencia (QoE) se puede entender como una medida de **extremo a extremo** del **rendimiento y prestaciones** del sistema de comunicaciones, siempre desde el punto de vista y desde la **perspectiva del usuario**, influenciada por las expectativas del mismo, indicando el **grado de satisfacción**.

2.3.1 Medida de la Calidad Percibida en Servicios de Telecomunicación

Existen diferentes alternativas y métodos para medir o evaluar la calidad de experiencia en el servicio IPTV [12], pudiendo dividirlos en: subjetivos, objetivos e indirectos. [16]

Métodos subjetivos

El receptor final de cualquier sistema de vídeo es el ser humano, por lo que el objetivo final es satisfacer al espectador con alta calidad de imagen. Por ello, una evaluación subjetiva empleando espectadores humanos es una de las técnicas más precisas de la calidad de vídeo.

Las evaluaciones visuales de la calidad de vídeo dependen de múltiples factores y parámetros, tales como la distancia de visualización, el tamaño de la pantalla y resolución, brillo, contraste, sombras, color, etc. Debido a esto, las evaluaciones subjetivas formales se realizan mediante métodos y procedimientos concretos, descritos y detallados por la ITU-T P.800, en la que se establecen las pautas sobre cómo definir los cuestionarios que deben completar los espectadores encuestados, así como los entornos y vídeos a mostrar. La escala de medición empleada es el MOS (*Mean Opinion Score*).

Cada usuario/espectador otorga una calificación numérica al grado de satisfacción de la visualización del vídeo, englobando dicha puntuación todos los aspectos, tanto objetivos como subjetivos, que influyen en la satisfacción del espectador. La escala MOS va desde 1 (mala calidad) hasta 5 (calidad excelente), tal y como se muestra en la tabla 2.1.

Tabla 2.1. Escala MOS (*Mean Opinion Score*)

| MOS | Calidad | Deterioro |
|-----|------------|-----------------------------|
| 5 | Excelente | Imperceptible o inexistente |
| 4 | Buena | Perceptible pero no molesto |
| 3 | Razonable | Ligeramente molesto |
| 2 | Deficiente | Molesto |
| 1 | Muy mala | Muy molesto |

La puntuación MOS final es el resultado de la media aritmética de las calificaciones de todos los espectadores y todas las secuencias de vídeo propuestas.

Se trata de un método muy fiable, utilizado también para medir la calidad en otros servicios multimedia (VoIP, Voz sobre IP), aunque presenta limitaciones e inconvenientes, principalmente el elevado coste de recursos y tiempo necesarios para su elaboración, no pudiendo ser empleado en mediciones de tiempo real, así como la limitación de no poder distinguir los aspectos que determinan la baja calidad percibida. No obstante, es un método que se utiliza como *ground truth* para comprobar la precisión de otros métodos.

Métodos objetivos

Los métodos objetivos son capaces de obtener resultados de forma más económica, con un mayor rendimiento, siendo posible utilizarlos en herramientas de monitorización. Los métodos objetivos de estimación de la Calidad de Experiencia [3] admiten una doble clasificación: en función de la información necesaria para la medición y en función del tipo de análisis que se realiza al vídeo.

Fijándonos en la *cantidad de información* que requieren, se distingue:

- Referencia completa (FR, *Full Reference*): método en el cual tanto la señal original transmitida como la señal recibida están disponibles para determinar la calidad de vídeo. Se trata del grupo de algoritmos más preciso, otorgando buenos resultados, pero son muy costosos y lentos computacionalmente.
- Referencia reducida (RR, *Reduced Reference*): método en el que se dispone de información parcial de la señal de vídeo transmitida, junto con la señal completa recibida, empleadas para determinar la calidad de vídeo. Se trata de un método más ligero y económico, tanto en tiempo como en recursos, aunque proporciona resultados menos precisos.
- Sin referencia (NR, *Non Reference*): método en el que únicamente se dispone de la señal de vídeo recibida. Se trata del mejor método para aplicaciones en tiempo real, aunque con resultados peores.

La clasificación que atiende al *tipo de análisis* diferencia:

- Medidas basadas en datos: se encargan de analizar, píxel a píxel, los cambios entre la imagen recibida y la imagen transmitida de referencia. El algoritmo más conocido dentro de este esquema es PSNR (*Peak Signal to Noise Ratio*). PSNR, Relación Señal a Ruido de Pico, se computa utilizando el valor medio del MSE (*Mean Square Error*) de cada imagen (*frame*).
- Medidas basadas en imagen, para las que existen dos enfoques:

- Enfoque de modelado de la visión (psicológico): trata de analizar varios componentes del sistema visual humano (HVS), como contraste, percepción del color, enmascaramiento temporal y espacial, etc.
- Enfoque de ingeniería: se imita el comportamiento del sistema visual humano. Son procesos muy costosos con algorítmicas muy complejas. Se encuentran aquí los siguientes métodos:
 - SSIM (*Structural SIMilarity*): el valor SSIM se calcula para cada *frame* y se realiza la media de toda la secuencia de vídeo. Mide la similitud de las dos imágenes por ventanas de píxeles. Se suele aplicar únicamente a la componente de luminancia. Devuelve un valor entre -1 y 1, correspondiendo 1 a dos imágenes idénticas.
 - MS-SSIM (*MultiScale SSIM*): variante avanzada del método SSIM, en el que se calcula el SSIM múltiples veces, con diferente escala (sub-muestreo) de la imagen.
 - Speed-SSIM.
 - VQM (*Video Quality Metric*). Se explica con más detalle en el punto siguiente. [9]
 - MOVIE (*Motion based Video Integrity Evaluation*). Emplea modelos del procesamiento de señales por parte del cerebro humano. Estima la calidad percibida mediante información espacial y temporal.
 - VSNR (*Visual Signal to Noise Ratio*). Se computa utilizando el total de la señal y el ruido de la secuencia de vídeo.

La desventaja de estas medidas es la necesidad de tener la imagen de referencia (la señal original) sincronizada con la señal recibida, por lo que se hace complejo de utilizar en aplicaciones de tiempo real.

- Medidas basadas en paquetes (flujo de bits): analizan la traza de vídeo enviada por la red de paquetes IP, mediante el análisis de la cabecera de los mismos. Tienen un bajo coste computacional, permitiendo un análisis simultáneo en red de múltiples canales.

Métodos indirectos

Se trata de métodos que relacionan parámetros de calidad de servicio, QoS (retardo, jitter, pérdida de paquetes, etc.), con un método subjetivo u objetivo, para medir la calidad de experiencia (QoE). Estos métodos precisan de la generación de un modelado matemático, mediante regresión multivariante, para obtener un modelo de correlación de los diferentes parámetros de la calidad de servicio con la calidad de experiencia. Con dicho modelo es posible predecir la calidad de experiencia del vídeo.

De esta manera, la entrada del sistema la constituyen los parámetros de calidad de servicio, con los que, una vez aplicado el modelo de correlación, se obtiene la medida de la calidad de experiencia, tal y como se muestra en la figura 2.19.

Este Trabajo Final de Máster se centrará en la generación de un sistema similar, en el que los parámetros de calidad de servicio serán la tasa de pérdidas de paquetes IP.

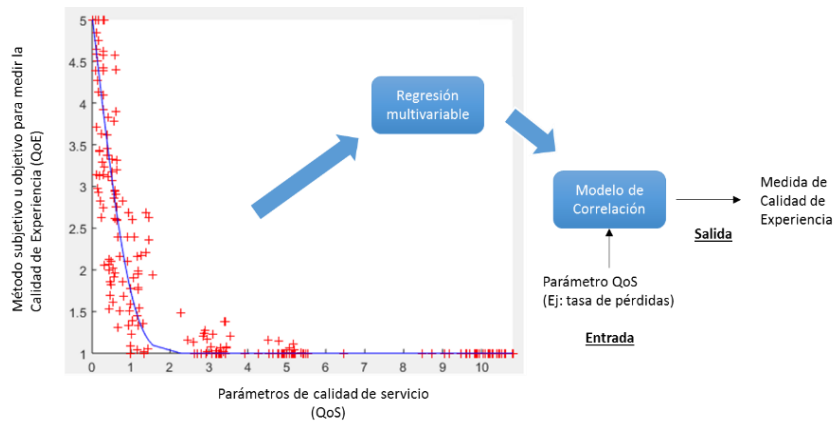


Figura 2.19: Proceso de medición Calidad de Experiencia en IPTV con Método Indirecto

2.3.1.1 VQM

VQM (*Video Quality Metric*) [11] es un método estándar de estimación de Calidad de Experiencia desarrollado por la NTIA (*National Telecommunications and Information Administration*). Se trata de un método de medida objetivo *Full-Reference* (FR) con resultados muy cercanos a los valores obtenidos con métodos subjetivos.

Debido a los excelentes resultados obtenidos en las pruebas de validación llevadas a cabo por el grupo VQEG (*Video Quality Expert's Group*), el método VQM fue adoptado por los estándares de la ITU y ANSI (*American National Standards Institute*).

El proceso necesario para la obtención del parámetro VQM se puede ver en la figura 2.20. Al tratarse de un algoritmo *Full-Reference*, hace uso de la imagen de referencia, de la cual son extraídas características para compararlas con la imagen procesada (transmitida por la red).

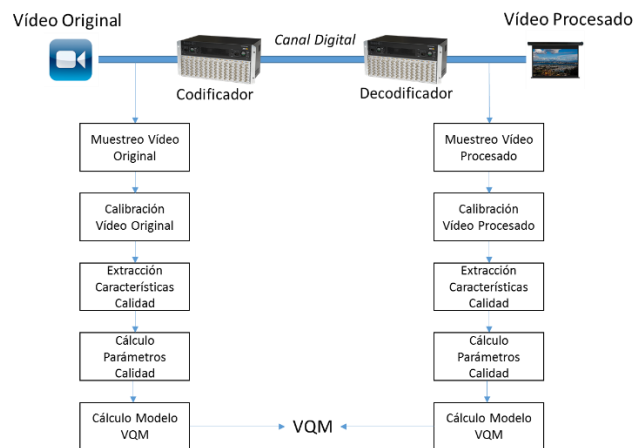


Figura 2.20: Proceso de cómputo VQM

El modelo general VQM obtiene resultados muy próximos a los obtenidos con un método subjetivo, con un coeficiente de correlación próximo a 1. En la figura 2.21 vemos que el coeficiente de determinación es 0,948.

La fórmula que ajusta los datos VQM a la escala MOS (Calidad Subjetiva) es la mostrada en (2.1).

$$MOS = 5 - 4 \cdot VQM \quad (2.1)$$

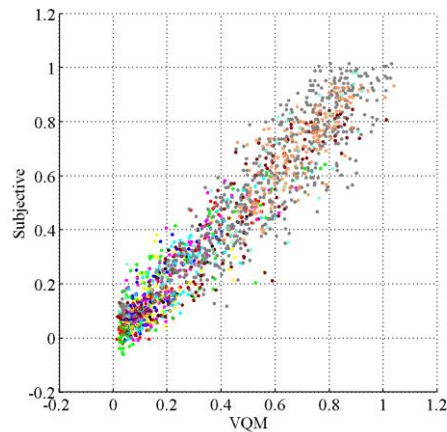


Figura 2.21: VQM frente a Calidad Subjetiva [11]

2.4 Conclusiones

En este capítulo de la memoria se ha mostrado la base tecnológica del servicio IPTV, detallando los estándares de codificación y transmisión que se emplean para el caso concreto de la televisión de alta definición (HDTV). Se han explicado las diferentes alternativas de encapsulación IPTV, detallando las ventajas e inconvenientes de cada una de ellas.

Asimismo, una vez concretado y aclarado el concepto de “Calidad Percibida”, se han mostrado los diferentes métodos existentes de medida de la misma, comparando rendimiento y resultados de cada uno, y describiendo el método a seguir para la construcción de la herramienta de monitorización.

En capítulos posteriores se tomarán las ideas expuestas en éste con el fin de resolver el problema propuesto en los objetivos del Trabajo Final de Máster, realizando un análisis en profundidad del caso y mostrando el desarrollo de la solución, siguiendo los algoritmos y métodos elegidos, aquí explicados.

3. Análisis del problema

3.1 Introducción

En este apartado de la memoria se aborda el problema tratado en este Trabajo Final de Máster: la monitorización en tiempo real de la calidad de experiencia en sistemas de IPTV para imagen en alta definición, desde el aspecto técnico.

Una vez expuestos, de forma teórica, los trabajos previos y estudios relacionados con esta problemática, se decide continuar en la línea de los **métodos indirectos**, proponiendo una solución capaz de tomar medidas de la red IP y relacionarlas, mediante un modelado matemático, con la calidad de experiencia del espectador.

La solución propuesta, por tanto, se centra en la codificación de vídeo H.264 (alta definición), junto con el estándar de transmisión MPEG-TS, analizando las pérdidas tanto a nivel de Unidad NAL (H.264) como a nivel TS (*Transport Stream*). Para este fin, se entrenará al modelo con el método *Full-Reference* VQM, debido a la buena correlación que tiene con la calidad subjetiva (MOS) y por ser elegido como estándar de medición de calidad percibida por la ITU y ANSI.

Asimismo, dicha solución debe cumplir con la premisa de funcionar en tiempo real, con el objetivo de ser utilizada como herramienta de monitorización. Por ello, la solución elegida es la alternativa de menor coste computacional posible, sin necesidad de ningún tipo de referencia de la imagen original y sin necesidad de decodificar en ningún momento la imagen recibida.

3.2 Análisis de requisitos

En este apartado se especifican y listan los requisitos, tanto funcionales como no funcionales, que deberá cumplir la solución desarrollada para garantizar la consecución de los objetivos marcados de forma exitosa.

3.2.1 Requisitos funcionales

En esta sección se especifican los requerimientos de carácter funcional que deberá cumplir la herramienta implementada, es decir, las funcionalidades y comportamientos que tendrá el sistema ante diferentes casos de uso del mismo.

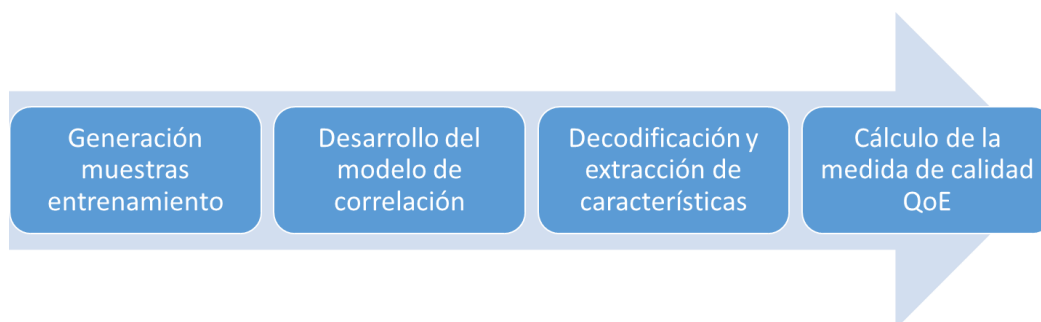


Figura 3.1. Diagrama de contexto desarrollo de la herramienta

Los requisitos funcionales se subdividen por módulos del sistema (figura 3.1), cada uno de los cuales se encarga de realizar una serie de tareas dentro del sistema de monitorización

global. Los módulos se corresponden directamente con los objetivos propuestos en este Trabajo Final de Máster:

- **Generación de las muestras de entrenamiento:** módulo encargado de obtener y generar todas las secuencias de vídeo de entrenamiento, con diferentes tasas de *bit rate*, y con diferentes degradaciones, así como extraer las características de tasa de pérdidas, junto con sus calidades subjetivas (VQM).
- **Desarrollo del modelo de correlación:** este módulo deberá realizar un análisis de los datos generados por el módulo anterior (QoS = PLR vs QoE = VQM), y obtener la correlación existente.
- **Decodificación y extracción de características:** tendrá la tarea de leer las capturas de tráfico de las secuencias de vídeo de test y extraer las tasas de pérdidas (PLR, *Packet Loss Rate*)
- **Cálculo de la medida QoE:** será el módulo encargado de aplicar la fórmula de correlación del modelo matemático a las estadísticas de tasas de pérdidas, obteniendo como salida la medición de calidad de experiencia (QoE).

Seguidamente se muestran los requisitos funcionales para cada uno de los módulos explicados arriba:

Generación de las muestras de entrenamiento

RF1: Se deberá disponer de un número mínimo de secuencias de vídeo (aprox. 15) que combinen diferentes tipos de vídeo (movimiento, cambios de escena, iluminación, etc.) y diferentes parámetros de tasa binaria.

RF2: Se deberá disponer del mismo vídeo con diferentes tasas de degradación, repitiéndose el proceso un número de veces para obtener un subconjunto de muestras lo suficientemente significativo.

RF3: Se almacenarán en capturas de tráfico *pcap* todas las retransmisiones realizadas para generar las muestras de entrenamiento.

RF4: Deberá ser capaz de obtener las estadísticas de calidad de servicio, mediante la referencia de la captura de tráfico original.

RF4.1: Se extraerán características de tasas de pérdidas por paquetes IP (PLR).

RF4.2: Se extraerán características de tasas de pérdidas por tipo de imagen MPEG (I, P, B).

RF5: Las secuencias de vídeo deberán cumplir una serie de requisitos en cuanto a formato y duración.

RF5.1: El formato de las secuencias de vídeo deberá ser un formato *RAW YUV 4:2:2*, compatible con la herramienta utilizada VQMT.

RF5.2: La duración máxima de las secuencias de vídeo deberá ser de 15 segundos.

RF6: Se realizarán dos subconjuntos de datos: H.264 sobre MPEG-TS y RTP Nativo.

Desarrollo del modelo de correlación

RF7: Se deberán extraer medidas objetivas sobre el grado de ajuste del modelo (Coeficiente de determinación y Error cuadrático medio).

RF8: No existirá ninguna limitación en cuanto al tipo de regresión realizada (lineal o no lineal).

RF9: Se realizará el ajuste de los datos de forma global (PLR vs QoS) y de forma independiente para cada tipo de imagen (IL, *I Loss*, vs QoS; PL, *P Loss*, vs QoS; BL, *B Loss*, vs QoS).

RF10: Se realizarán dos modelos independientes: H.264 sobre RTP y H.264 sobre MPEG-TS.

Decodificación y extracción de características

RF11: Se analizarán las capturas de red (*pcap*) a nivel de RTP/NALU (número de secuencia, *timestamp*, etc.).

RF12: Se analizarán las capturas a nivel de MPEG (tipo de cuadro: I, P B), o directamente a nivel de H.264 (*slice type* I, P, B).

RF13: Se obtendrán estadísticas de calidad de servicio: tasa de pérdidas sin ninguna referencia.

RF13.1: Se obtendrán tasas de pérdidas de paquetes (PLR).

RF13.2: Se obtendrán tasas de pérdidas por tipo de cuadro (IL, PL, BL).

RF14: Se deberá parametrizar el código para facilitar la configuración.

Cálculo de la medida QoE

RF15: Se estimará el valor de Calidad de Experiencia (QoE) con los datos de la red.

RF16: Se almacenarán los valores de calidad en un fichero de texto, de forma periódica.

RF17: Se deberá parametrizar el código para facilitar la configuración.

3.2.2 Requisitos no funcionales

Por otro lado, la solución desarrollada para la monitorización también deberá cumplir con otra serie de criterios no relacionados con su funcionalidad. Se trata de características que garantizan la robustez y fiabilidad de la herramienta:

RNF1: Rendimiento. La herramienta deberá tener un alto rendimiento, para cumplir el objetivo de funcionar en tiempo real, así como para integrarse en dispositivos de bajo coste.

RNF2: Disponibilidad. La herramienta deberá estar siempre disponible para monitorizar de modo constante el tráfico y poder detectar y solucionar problemas en la red.

RNF3: Estabilidad. La aplicación deberá ser muy estable, sin caídas ni fallos, ya que estará en proceso continuo de monitorización.

RNF4: Portabilidad. La aplicación será desarrollada en código abierto y podrá ejecutarse en cualquier sistema Linux.

RNF5: Mantenibilidad. El código fuente y scripts de la herramienta deberá estar estructurado y organizado para facilitar cualquier cambio, así como el mantenimiento de la aplicación.

RNF6: Gestionabilidad. El código debe estar correctamente comentado y documentado.

RNF7: Seguridad. Los datos almacenados de las capturas de tráfico deberán estar en un entorno protegido.

RNF8: Usabilidad. Tanto el uso de la herramienta como los datos generados por ésta deben ser sencillos y fáciles de comprender.

3.3 Conclusiones

En este capítulo se han especificado las características y criterios a cumplir por la herramienta de monitorización. Dichos requerimientos se han separado en módulos, con el objetivo de facilitar la comprensión, el análisis, el mantenimiento y el cumplimiento de los mismos. Se hace especial hincapié en la generación de una herramienta escalable, que permita su modificación y evolución de forma sencilla ante cambios futuros. También destaca la necesidad de generar una herramienta capaz de funcionar en tiempo real en sistemas de bajo coste.

En los capítulos siguientes se explicará el detalle de la construcción de la herramienta, mostrando las soluciones adoptadas para cada uno de los requisitos.

4. Desarrollo de la solución

4.1 Introducción

En este capítulo se muestra con detalle el proceso seguido para el diseño y construcción de la herramienta de monitorización.

Tal y como se indicó en el capítulo anterior, el desarrollo del sistema de monitorización está dividido en módulos (figura 3.1):

- **Generación de las muestras de entrenamiento**
- **Desarrollo del modelo de correlación**
- **Decodificación y extracción de características**
- **Cálculo de la medida de Calidad de Experiencia (QoE)**

En los siguientes apartados del capítulo se explicará, modulo a modulo, los detalles del proceso realizado en cada uno de ellos, indicando los programas, herramientas y otros recursos empleados, así como los resultados parciales obtenidos.

4.2 Generación de las muestras de entrenamiento

Para el desarrollo de la herramienta es necesario generar una base de datos de vídeos en alta definición [13], con diferentes tasas de imagen (*frame rate*) y con diferentes características espaciales y temporales, consiguiendo así muestras suficientemente heterogéneas para considerar válido el modelo.

Para ello, se han empleado un total de 20 secuencias de vídeo diferentes, todos ellos en formato sin compresión YUV4MPEG, formato de vídeo 1080p (1920 x 1080 píxeles), extraídos de [15]. Se trata de un conjunto de vídeos altamente utilizados en experimentos similares.

Los pasos realizados para la generación de las muestras de entrenamiento se muestran en la figura 4.1.

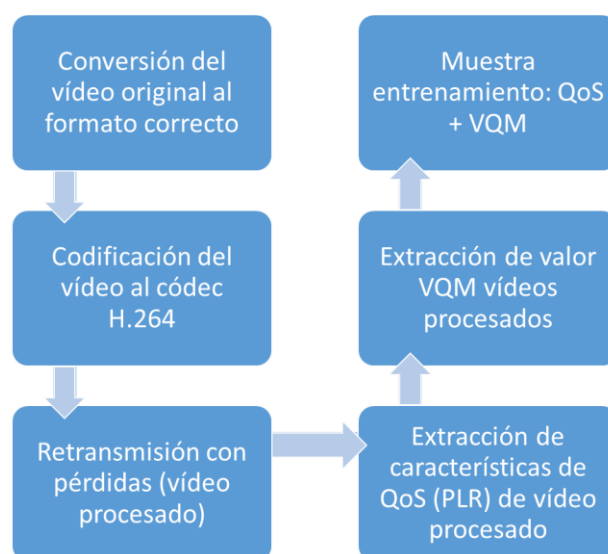


Figura 4.1. Diagrama Generación muestras de entrenamiento

Por cada uno de los vídeos de entrenamiento sin codificar se ha realizado una codificación en el códec H.264, mediante las dos encapsulaciones IPTV H.264 explicadas en el apartado 2.2.3: MPEG-TS sobre IP y RTP Nativo.

Tal y como se muestra en la figura 4.1, el primer paso a realizar es la conversión y adaptación al formato correcto de la secuencia de vídeo. El formato original obtenido en [14] es YUV4MPEG (Y4M), pero el formato aceptado por la herramienta VQMT (utilizada para obtener el valor de VQM) [11] es YUV 4:2:2 y con una duración máxima de 15 segundos.

A continuación, se realiza una codificación a H.264. Para ello, se hace uso de la herramienta *ffmpeg* [14]. Se trata de software libre que permite convertir (transcodificar) audio y vídeo, desarrollado en GNU/Linux.

Por cada vídeo se realiza la retransmisión por una red de área local con diferentes tasas de pérdidas (0,1%; 0,4%; 1%; 3%; 5% y 10%), mediante un modelo de estados Gilbert-Elliot [6][7] y con una ráfaga de pérdidas de 3 paquetes, característico de las redes IP [4] [22]. Este proceso se repite 10 veces por cada vídeo (5 veces para RTP y 5 veces para MPEG-TS). Con ello, el número total de muestras son 1200.

La herramienta utilizada para aplicar las pérdidas es *netem* [17], un módulo del kernel de Linux a través del cual se aplica el modelo de estados Gilbert-Elliot [7]. Se trata de un modelo de dos estados (Bueno (G, *Good*) y Malo (B, *Bad*)) y cuatro parámetros independientes (figura 4.2).

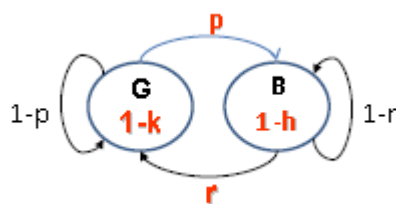


Figura 4.2. Modelo Gilbert-Elliot

El modelo aplicado en la práctica es el modelo Simple de Gilbert, el cual tiene dos estados, pero únicamente dos parámetros independientes, pudiendo ser posible modelar sistemas con pérdidas consecutivas, como el caso de las redes IP, caracterizados por una probabilidad de pérdidas (p) y una duración de ráfaga ($1/r$). En este caso, los otros dos parámetros son $h=0$ y $1-k=0$ (figura 4.3).

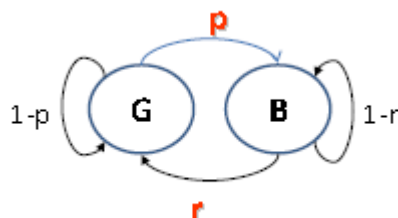


Figura 4.3. Modelo Simple de Gilbert

Tras ello, se obtienen los vídeos en H.264 degradados con diferentes tasas de pérdidas. De ello, es necesario extraer las características de calidad de señal (QoS), obteniendo las pérdidas reales (PLR, *Packet Loss Rate*), dado que la herramienta *netem* genera las pérdidas de forma aleatoria. Asimismo, se obtienen tasas de pérdidas en detalle por tipo de imagen H.264 (I, P, B).

Por último, con los vídeos YUV original y YUV degradados, mediante la herramienta VQMT se obtiene el valor VQM (indicativo de la calidad de experiencia, QoE). El algoritmo elegido, debido a su velocidad de procesamiento y sus correctos resultados, fue el algoritmo “Developers Model” el cual tiene un coeficiente de determinación de 0,973 frente a MOS (figura 4.4), ligeramente inferior al modelo General.

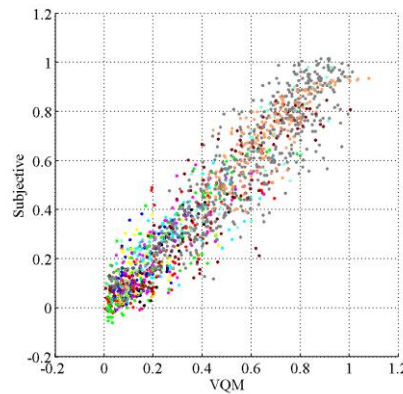


Figura 4.4. Algoritmo *Developers Model* VQM [11]

Tras esto se obtienen las **muestras de entrenamiento**, formadas por los datos de calidad de señal (QoS), pérdidas de paquetes, y datos de calidad de experiencia (QoE), VQM. Podemos ver en detalle los pasos del proceso completo en la figura 4.5,

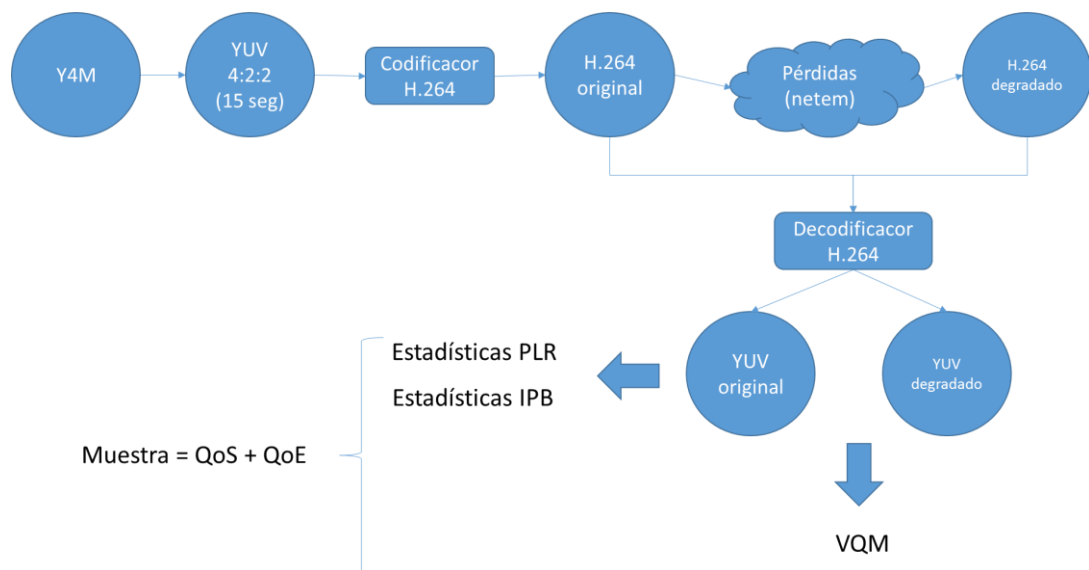


Figura 4.5. Detalle del Diagrama Generación de las muestras de entrenamiento

Con el fin de facilitar y acelerar el proceso de generación de las muestras de entrenamiento, se han desarrollado una serie de *scripts* que automatizan el proceso de generación de los vídeos degradados, así como de la extracción de las características de pérdidas, tanto a nivel de paquetes como a nivel de imagen I, P, B.

En la siguiente tabla (Tabla 4.1) se muestra un resumen de las características de la base de datos de vídeo HDTV con las que se han generado las muestras de entrenamiento:

Tabla 4.1. Base de datos vídeo HDTV

| <i>Base de datos vídeo HDTV</i> | |
|---------------------------------|--------------------------------------|
| <i>Número de secuencias</i> | 20 |
| <i>Resolución</i> | 1920x1080 (progresivo y entrelazado) |
| <i>Duración</i> | 10, 15, 19 segundos |
| <i>Frame rate</i> | 25, 30, 50 cuadros/s |
| <i>Tasas de pérdida</i> | 0,1%; 0,4%; 1%; 3%; 5%; 10% |

4.3 Desarrollo del modelo de correlación

El siguiente módulo del desarrollo es el ajuste de las muestras de entrenamiento a un modelo matemático, mediante una regresión.

Como primer paso al ajuste del modelo se visualizan las muestras, con el fin de observar la distribución de las mismas. Como se observa en la figura 4.6, los datos no siguen un comportamiento lineal. Por ello, se consideró realizar un ajuste no lineal, mediante funciones exponenciales e hiperbólicas. Este proceso se repite tanto para las muestras RTP Nativo como para las muestras encapsuladas con MPEG-TS, siguiendo ambas el mismo comportamiento.

4.3.1 Ajuste sobre PLR

En primer lugar, se realizó el ajuste de las muestras de entrenamiento en base a la pérdida de paquetes (PLR, *Packet Loss Rate*), paquetes RTP para el caso de RTP Nativo y paquetes TS para el caso de MPEG-TS.

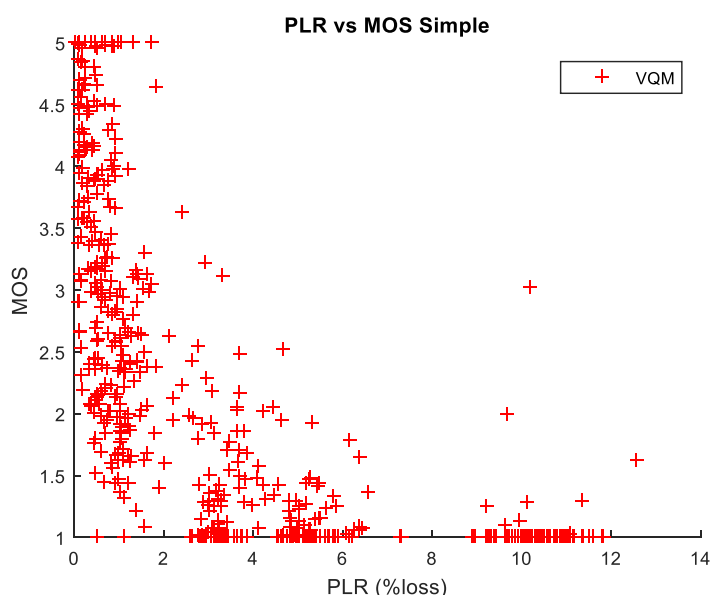


Figura 4.6. Dispersión VQM muestras entrenamiento (PLR Simple)

Como se mencionó anteriormente, las funciones utilizadas para el ajuste fueron la función exponencial y las funciones hiperbólicas. Los parámetros utilizados para medir la bondad del ajuste son el coeficiente de determinación (R^2) y el error cuadrático medio (SEE). Los resultados de dichos ajustes se observan en la figura 4.7:

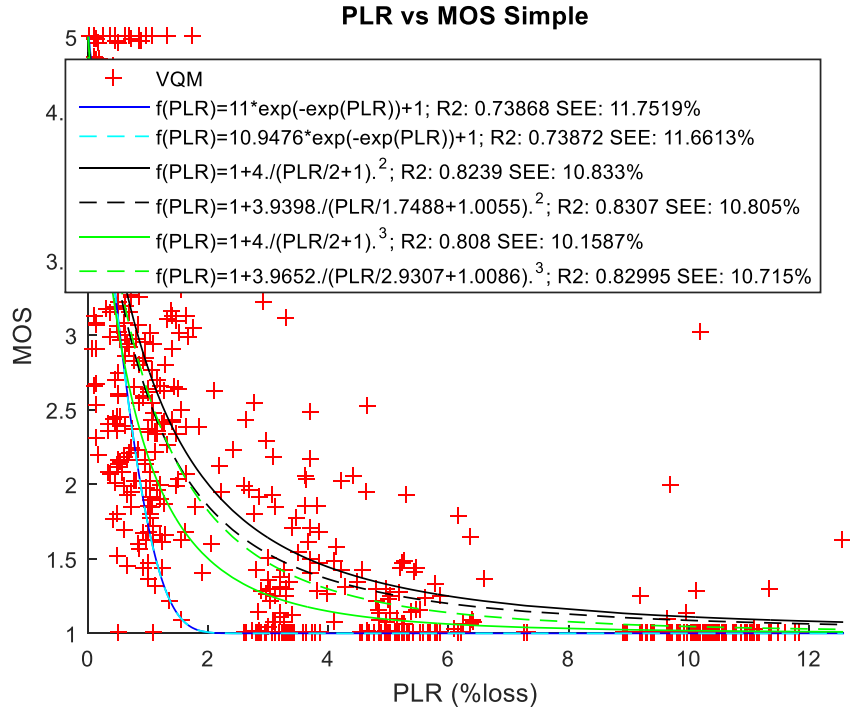


Figura 4.7. Ajuste muestras entrenamiento PLR Simple RTP

La función que ofrece mejor ajuste (mayor R^2) es la función hiperbólica cuadrada ($R^2 = 0,8307$). No obstante, la función con menor error es la función hiperbólica cúbica (SEE = 10,1587 %). En el apartado de validación de resultados, teniendo en cuenta las muestras de validación, se determinará la mejor función a utilizar en la herramienta.

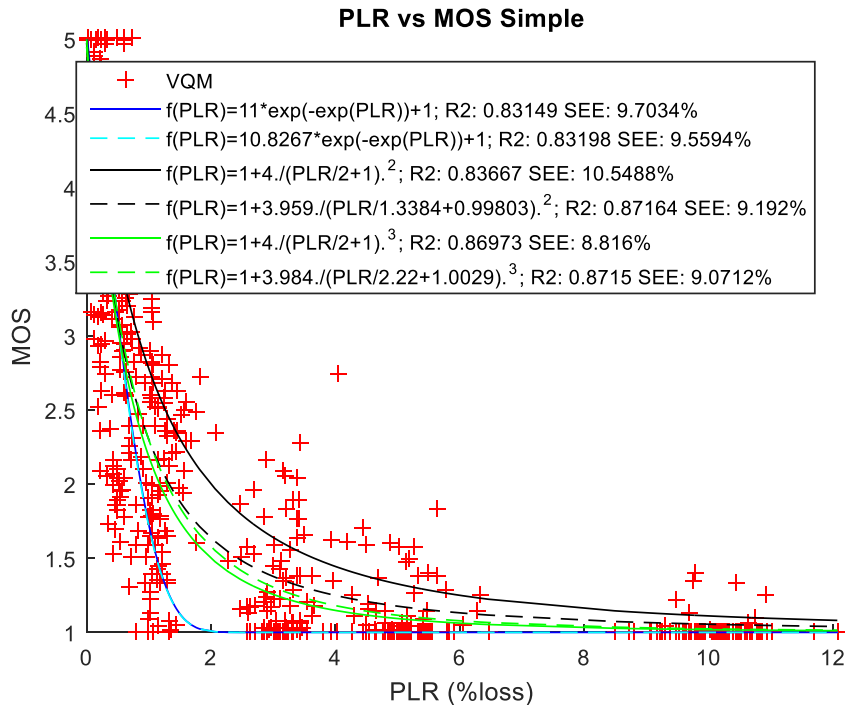


Figura 4.8. Ajuste muestras entrenamiento PLR Simple MPEG-TS

4.3.2 Ajuste sobre IPB

Como se explicó en el apartado 2.2.1, el estándar H.264 tiene distintos tipos de imágenes (I, P, B), las cuales son codificadas con distinta cantidad de información, siendo las imágenes I las únicas que no contienen ningún tipo de predicción, y las imágenes P y B las que están formadas con información de otras imágenes. Por ello, es obvio que pérdidas de paquetes con información de imágenes I tendrá mayor repercusión en la decodificación del vídeo, puesto que esa información forma parte de varias imágenes, obteniéndose vídeos con mayor degradación. Asimismo, y conociendo el funcionamiento de los decodificadores, se analiza la importancia de las cabeceras de las imágenes (el primer paquete de cada imagen [1]).

Por todo lo explicado, se realizó un análisis de las secuencias de vídeo en las que se diferenciaban las pérdidas por IL (Pérdidas imágenes I, *I Loss*), PL (*P Loss*) y BL (*B Loss*). De esta manera, las muestras de entrenamiento pasan a tener 4 dimensiones (IL, PL, BL y VQM).

En primer lugar, y como se realizó con el análisis PLR Simple, el primer paso fue la visualización de la muestras de forma independiente, con el objetivo de analizar la tendencia de las mismas (figuras 4.9, 4.10 y 4.11).

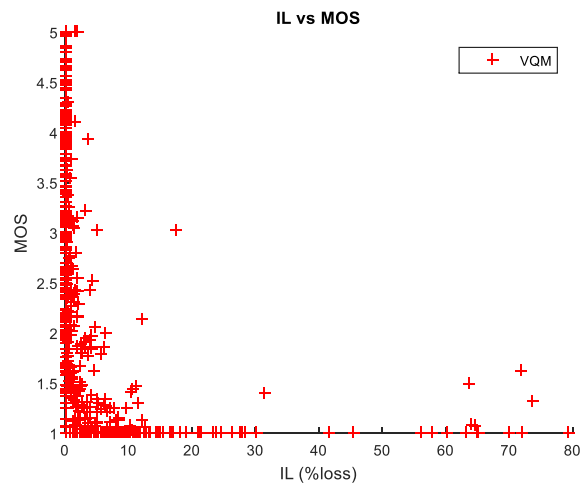


Figura 4.9. Dispersión muestras entrenamiento IL, RTP

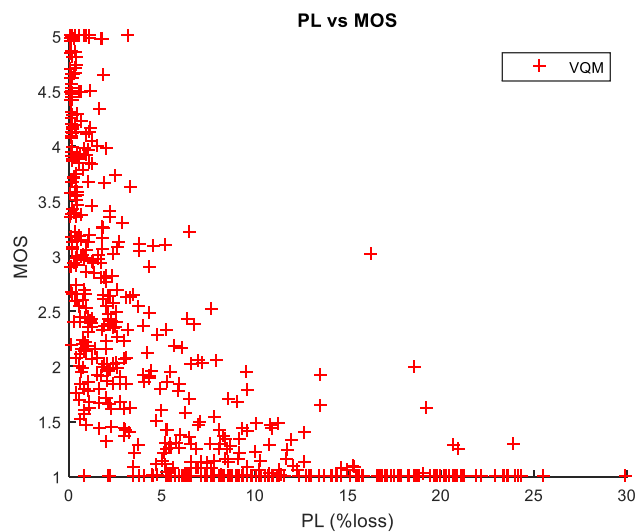


Figura 4.10. Dispersión muestras entrenamiento PL, RTP

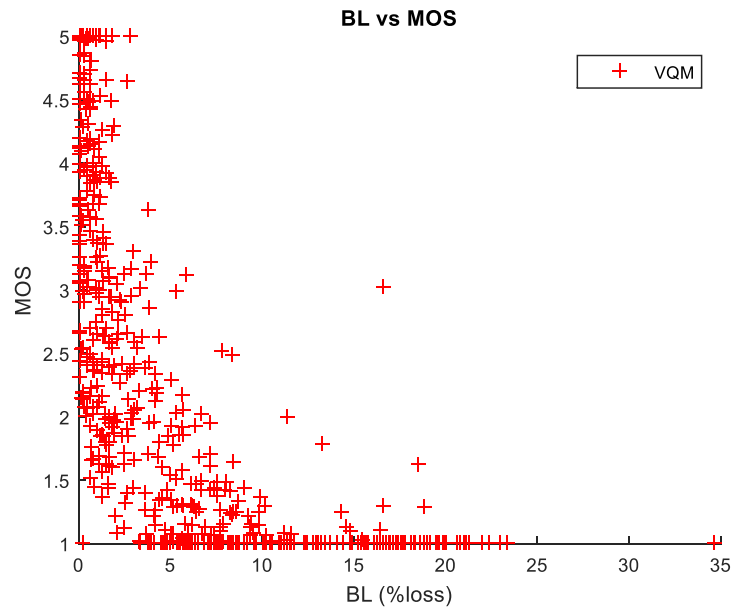


Figura 4.11. Dispersión muestras entrenamiento BL, RTP

Como se observa en las figuras anteriores, ninguna de las tres pérdidas siguen una distribución lineal, por lo que se pasó a analizar el ajuste mediante funciones exponenciales e hiperbólicas. Las distribuciones BL y PL son bastante similares, mientras que la distribución IL se muestra con una caída más abrupta, algo que puede estar relacionado con la explicación anterior, en la que se hacía incapié en la mayor influencia en la degradación de la calidad de vídeo las pérdidas de paquetes con información de imágenes I.

Este comportamiento es similar para las muestras de la encapsulación MPEG-TS (figuras 4.12, 4.13 y 4.14).

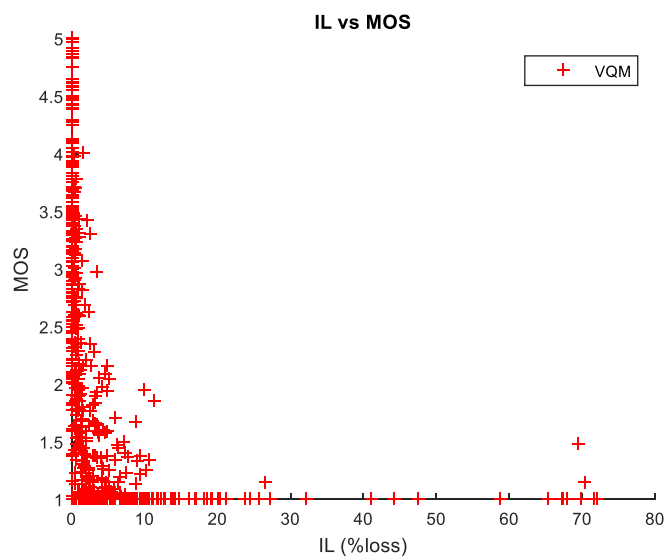


Figura 4.12. Dispersión muestras entrenamiento IL, MPEG-TS

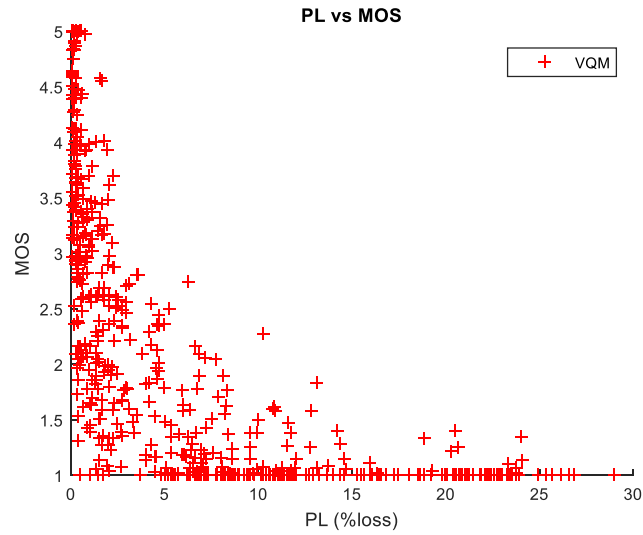


Figura 4.13. Dispersión muestras entrenamiento PL, MPEG-TS

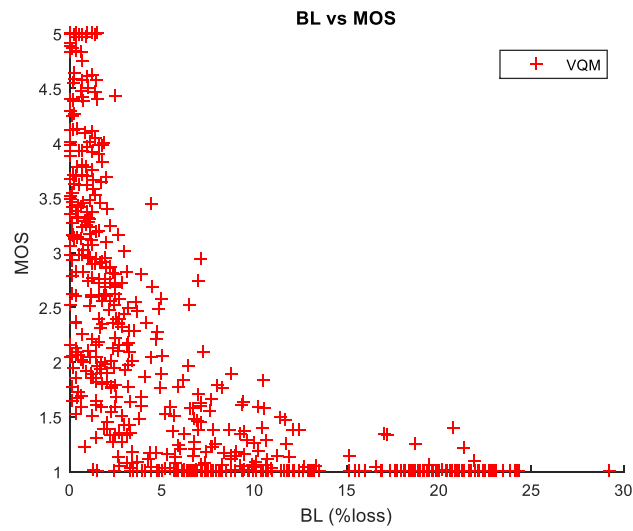


Figura 4.14. Dispersión muestras entrenamiento BL, MPEG-TS

En las siguientes figuras se muestran los resultados del ajuste de las muestras de entrenamiento teniendo en cuenta únicamente las pérdidas de cada uno de los tipos de imagen.

Para el caso del ajuste como función únicamente de las pérdidas de paquetes con información de imágenes I (IL), figura 4.15, la función con mejor ajuste es la exponencial ($R^2 = 0,69387$ y $SEE = 16,6569\%$).

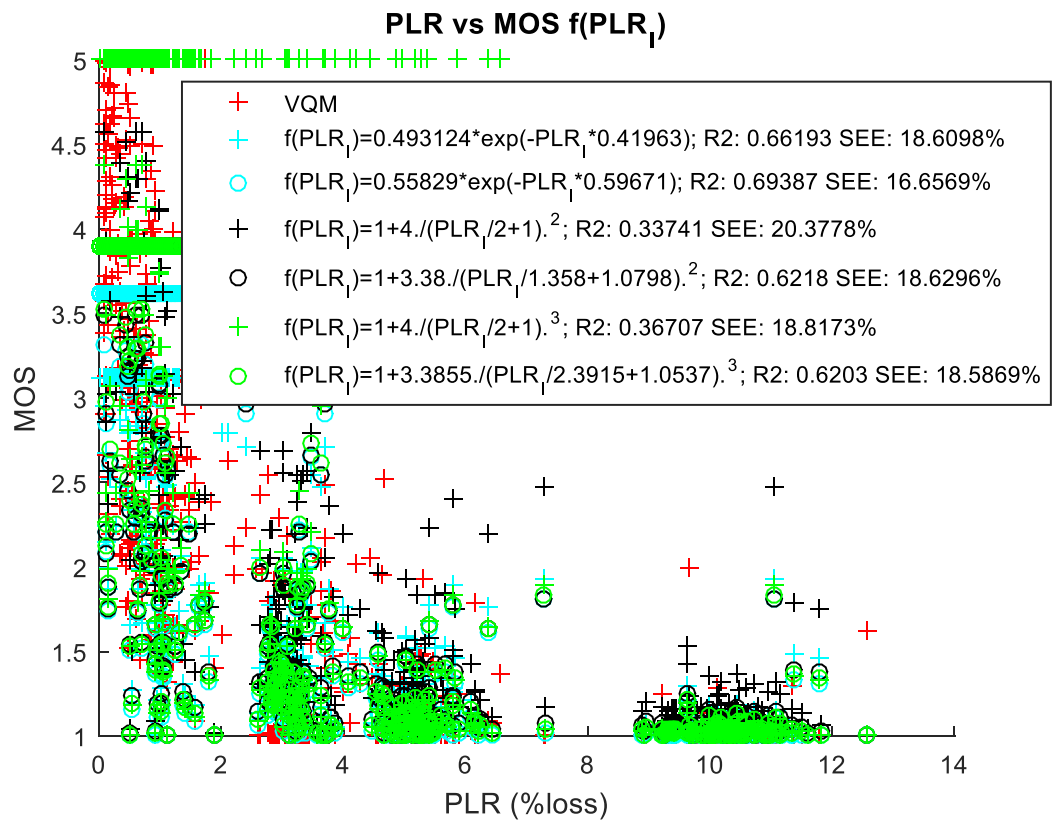


Figura 4.15 Ajuste muestras entrenamiento en función IL, RTP

En el caso del ajuste como función de PL, la función que proporciona un mejor ajuste (figura 4.16) es la exponencial ($R^2 = 0,83318$) y la función hiperbólica cuadrada la que ofrece menor error (SEE = 10,0363 %).

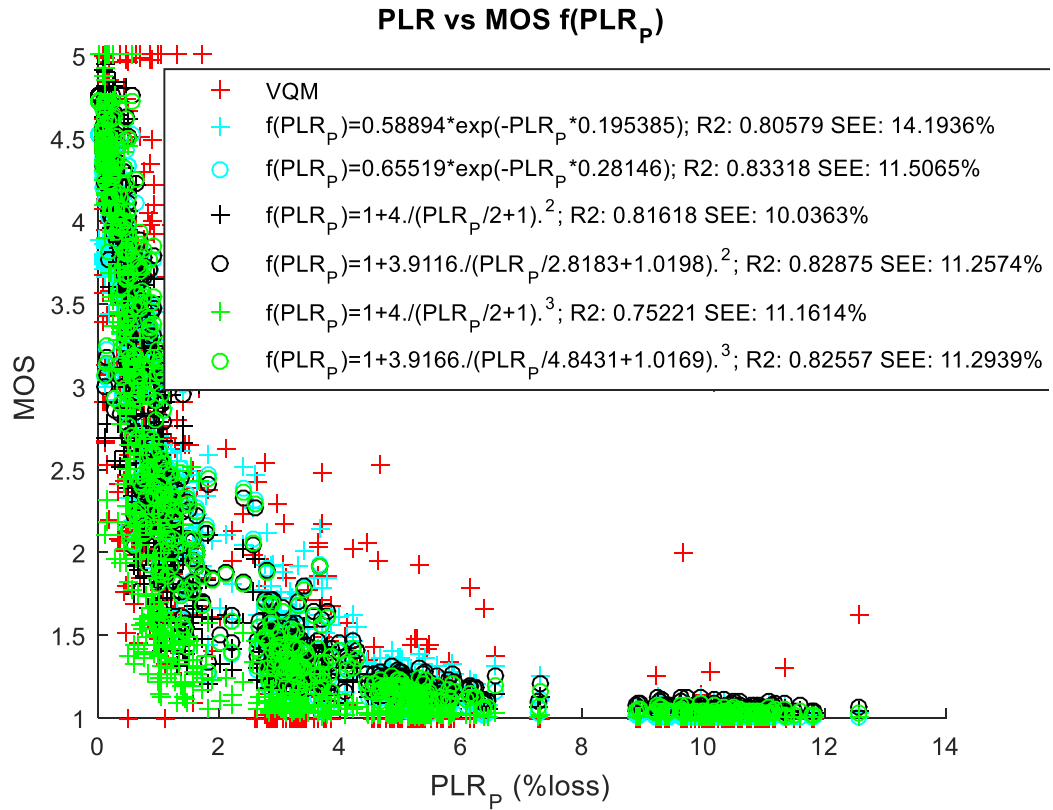


Figura 4.16. Ajuste muestras entrenamiento en función PL, RTP

Para el ajuste en función de BL (figura 4.17), la función con mejor ajuste es la exponencial ($R^2 = 0,81549$) y la función hiperbólica cuadrada (SEE = 11,7644 %).

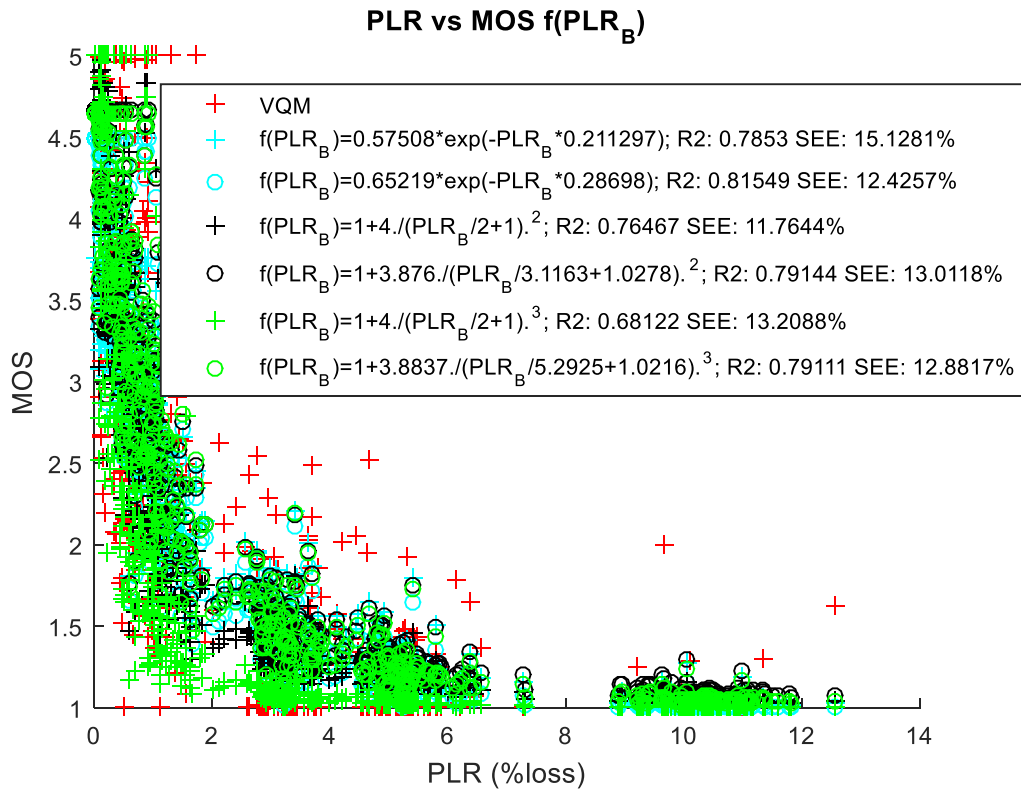


Figura 4.17. Ajuste muestras entrenamiento en función BL, RTP

De forma similar, se muestran los ajustes del modelo para el caso de la encapsulación MPEG-TS.

Para el ajuste en función de IL (figura 4.18), la función con mejor ajuste es la exponencial ($R^2 = 0,76312$ y $SEE = 14,0094\%$).

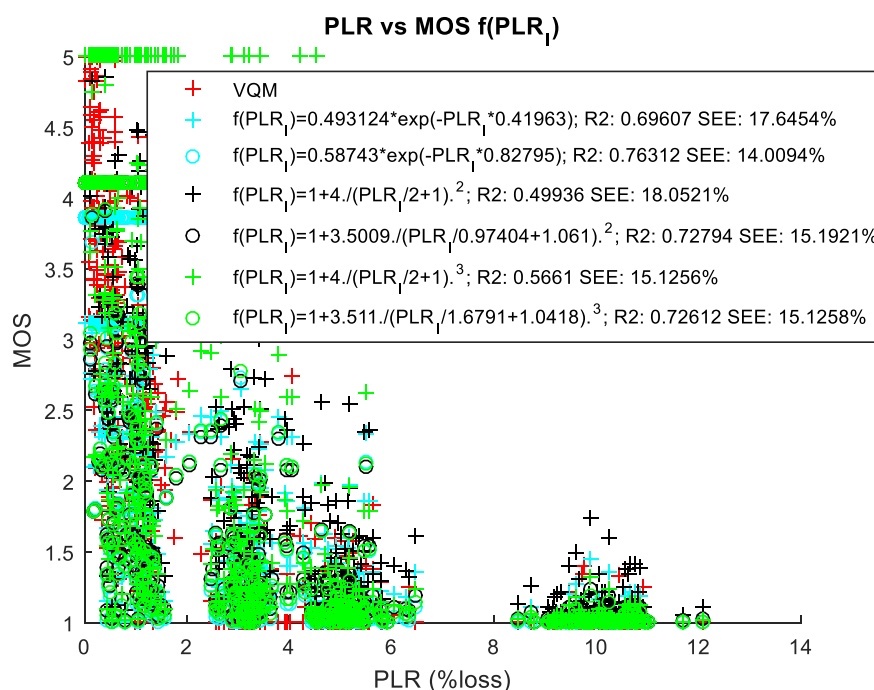


Figura 4.18. Ajuste muestras entrenamiento en función IL, MPEG-TS

Para el ajuste en función de PL (figura 4.19), la función con mejor ajuste es la hiperbólica cuadrada ($R^2 = 0,85251$ y $SEE = 9,4419\%$).

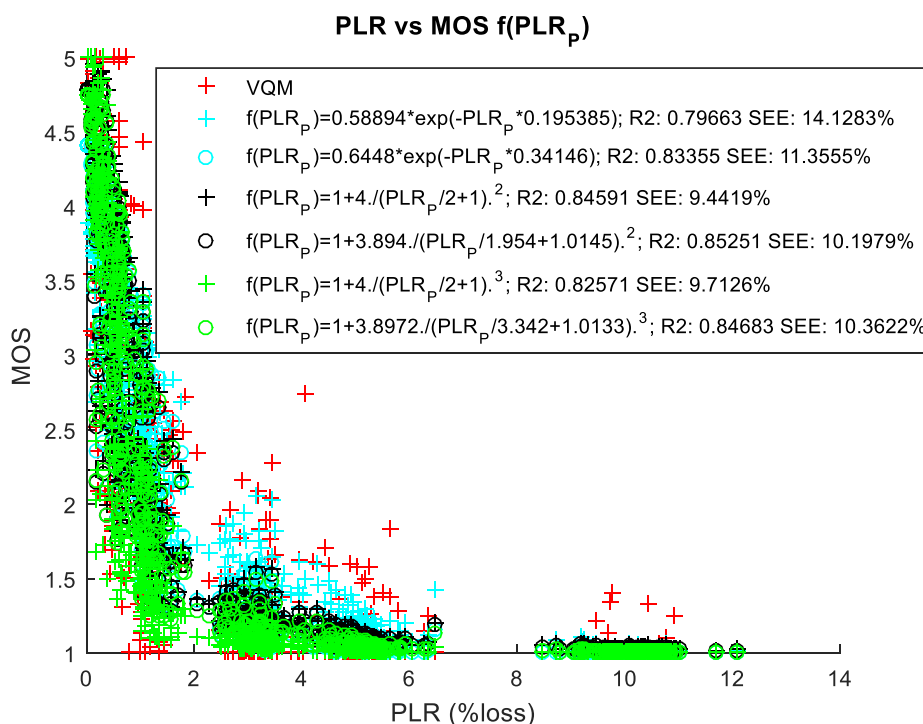


Figura 4.19. Ajuste muestras entrenamiento en función PL, MPEG-TS

Para el ajuste en función de BL (figura 4.20), la función con mejor ajuste es la exponencial ($R^2 = 0,81903$) y la función hiperbólica cuadrada (SEE = 11,2831 %).

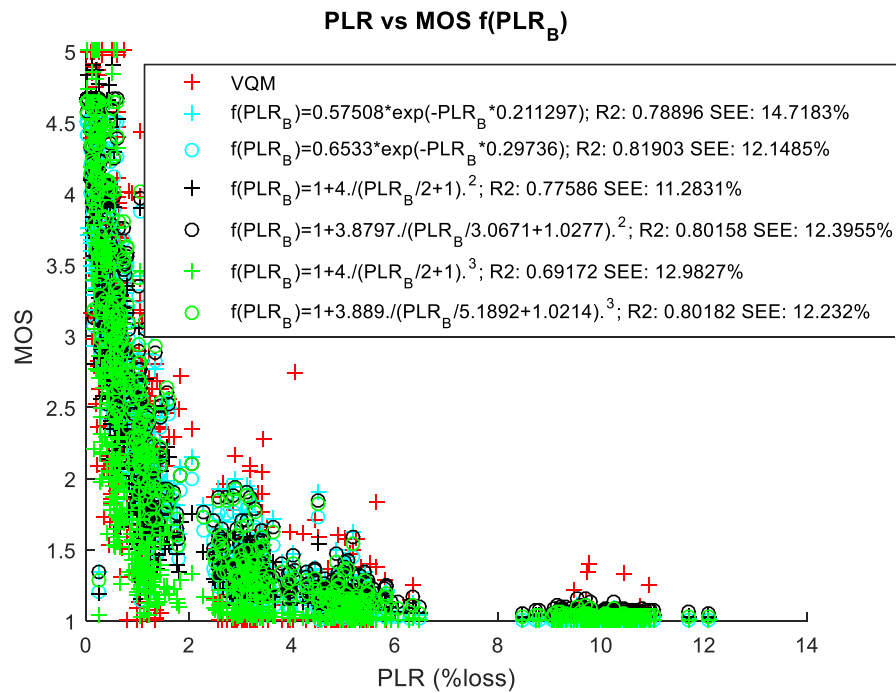


Figura 4.20. Ajuste muestras entrenamiento en función BL, MPEG-TS

Una vez visto los resultados del ajuste de forma independiente, se decidió combinar las 3 funciones: $f(IL, PL, BL)$ para ofrecer una estimación de la calidad subjetiva, es decir, realizamos una regresión multivariante (IL, PL, BL).

Los resultados se muestran en las figuras 4.21 y 4.22, donde podemos observar 4 funciones:

- Cian: combinación funciones exponenciales.
- Negro: combinación funciones hiperbólicas cuadradas.
- Verde: combinación de las funciones hiperbólicas cúbicas.
- Azul: combinación de las funciones con mejor ajuste para cada tipo de imagen (IL, PL, BL).

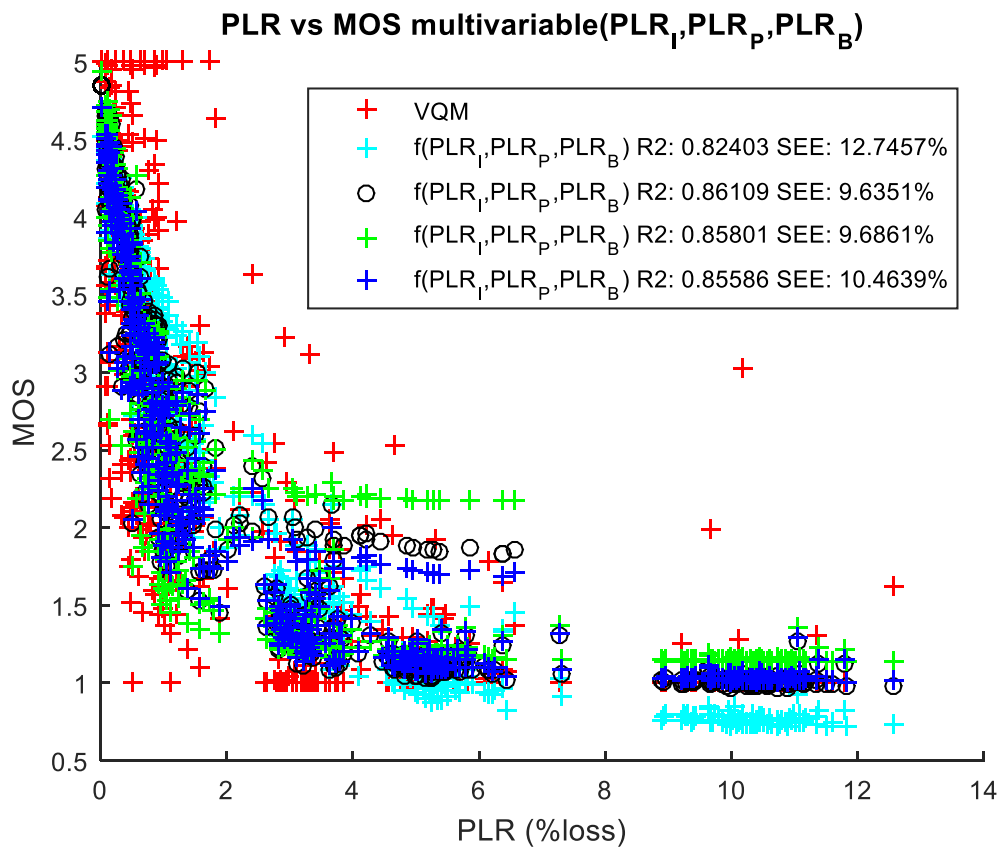


Figura 4.21. Ajuste muestras entrenamiento multivariante RTP

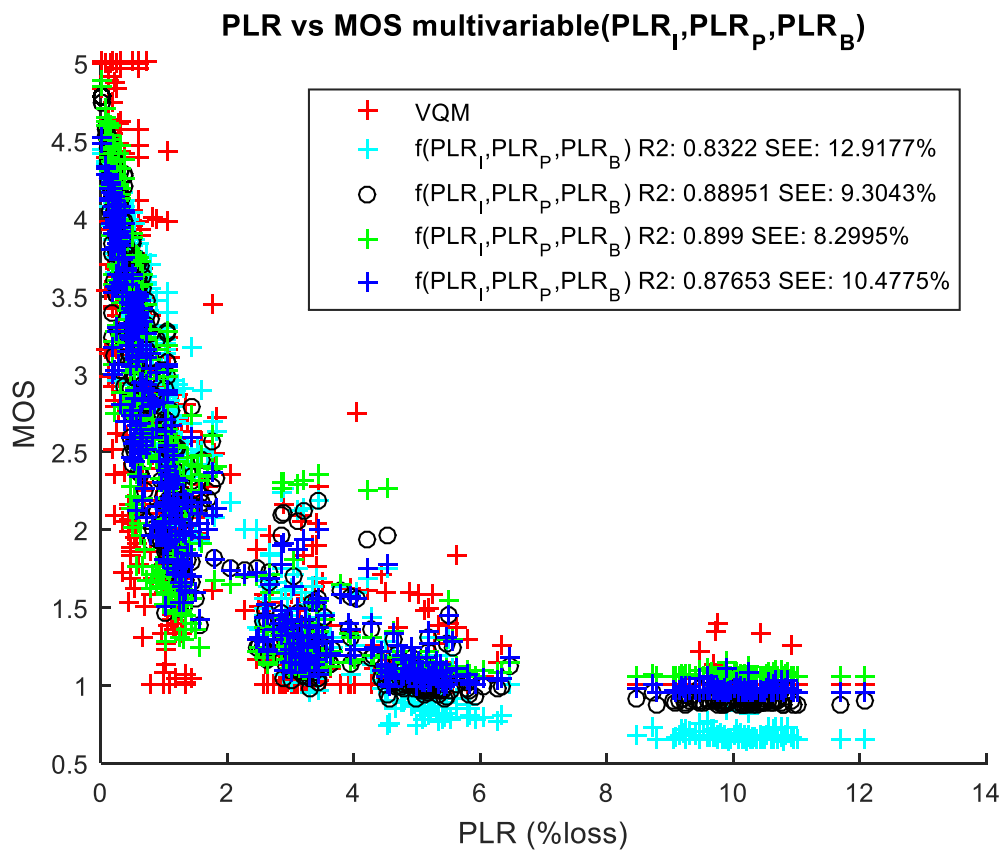


Figura 4.22. Ajuste muestras entrenamiento multivariante MPEG-TS

4.4 Decodificación y extracción de características

Una vez determinados los modelos matemáticos que ajustan a las muestras de entrenamiento, el siguiente paso para el desarrollo de la herramienta es implementar el decodificador de capturas *pcap* de secuencias de vídeo IPTV. Los pasos a seguir se muestran en la figura 4.23.



Figura 4.23. Diagrama de implementación del Decodificador capturas IPTV

Las secuencias de vídeo IPTV H.264 son capturadas en ficheros *pcap*, en las cuales están contenidos los paquetes de la transmisión del vídeo.

El decodificador y analizador de capturas es capaz de extraer la información contenida en las capturas, extrayendo información de parámetros de calidad de red, como la tasa de pérdida de paquetes. Asimismo, en un mayor nivel de análisis, es capaz de extraer información de vídeo H.264, permitiendo así proporcionar estadísticas de pérdidas a nivel de tipo de imagen: IL, PL y BL.

El programa se encuentra dividido en dos partes:

- Decodificador RTP Nativo: se encarga de analizar las capturas codificadas mediante encapsulación RTP/H.264, caso de Vídeo bajo demanda.
- Decodificador MPEG-TS: se encarga de analizar las capturas codificadas mediante el uso de MPEG-TS (Anexo A.1.2) como capa intermedia entre RTP y H.264.

Por ello, el programa tiene que ser capaz de distinguir entre capturas de ambos tipos, para lo que se recurre a la información del campo *RTP Payload Type*, siendo: 96 para el caso de RTP Nativo y 33 para el caso de MPEG-TS.

Ambos casos comparten conceptos: el fin del mismo es obtener **estadísticas de pérdidas**, tanto a nivel de paquetes RTP como de paquetes TS (en caso de MPEG-TS), así como a nivel de imagen H.264 (I, P, B). Por ello, el programa trata de detectar las pérdidas. Para ello, se utiliza la información presente en la cabecera RTP: Número de secuencia y Timestamp. Se considera como pérdida de paquete el caso en el que haya una pérdida de número de secuencia. Para el caso de MPEG-TS, una pérdida de un paquete RTP supone una pérdida de 7 paquetes TS (tal y como se explicó en el apartado 2.2.3.1).

Asimismo, para el desarrollo del decodificador y analizador de capturas se ha seguido la lógica de un decodificador H.264. Esto significa que se ha tenido en cuenta las pérdidas de las cabeceras de las imágenes, puesto que cuando un decodificador no recibe información de la cabecera de la imagen es incapaz de decodificar la imagen. Traducido a paquetes IP, se considera la cabecera de la imagen como el primer paquete IP que transporta información de la imagen nueva. De esta manera, cuando se detecta la pérdida de una cabecera, se desechan todos los paquetes que forman parte de dicha imagen. Para detectar este tipo de pérdidas, se tiene en consideración el valor del RTP Timestamp, sabiendo que todos los paquetes de una misma imagen llevan asociados el mismo Timestamp.

Resumiendo, podemos estar presentes ante dos tipos de pérdidas:

- Pérdida de un paquete interno de la imagen: se produce cuando hay una pérdida del número de secuencia RTP. Conlleva en la pérdida de información de algún macrobloque de la imagen.
- Pérdida de la cabecera de la imagen: se produce cuando se pierde el primer paquete RTP de la imagen. Se detecta cuando existe una pérdida del número de secuencia RTP y un cambio en el Timestamp.

Asimismo, para el caso de MPEG-TS, se hace uso del campo *CC Continuity Counter*, para mejorar las estadísticas de pérdidas TS.

Por otro lado, como se ha mencionado anteriormente, el programa tiene la capacidad de estimar pérdidas diferenciando por el tipo de imagen, es decir, de calcular IL (*I Loss*), PL (*P Loss*) y BL (*B Loss*). Para conseguir estas estadísticas se añade una mayor complejidad al proceso, puesto que se hace imprescindible conocer a qué imagen y de qué tipo pertenece cada paquete RTP (TS en caso de MPEG-TS).

Para conseguir este fin, es necesario analizar la carga útil de los paquetes, diferenciando el caso RTP Nativo y el caso MPEG-TS.

- RTP Nativo: en este caso, y como explicamos en el apartado 2, los paquetes RTP contienen unidades NAL. Dichas unidades NAL contienen dos octetos de cabecera (FU identifier y FU Header). En el caso del primer paquete de la imagen, el primer byte de la unidad NAL, nos indica qué tipo de *slice* está codificada:
 - 7: slice I.
 - 5: slice P.
 - 6: slice B.

En esta encapsulación, el último paquete que codifica información de la misma imagen viene marcado con un flag de fin (*End bit* en la cabecera FU Header).

- MPEG-TS: este caso es de mayor complejidad, puesto que es necesario analizar los distintos TS contenidos en cada paquete RTP, siendo necesario conocer:
 - Información PAT: nos proporciona el PID de la tabla PMT.
 - Información PMT: nos proporciona el PID del flujo de vídeo H.264. Una vez conocido, únicamente serán de interés los paquetes TS con PID = PID_264, puesto que son los paquetes que llevan la información del vídeo H.264.

Para detectar el tipo de imagen codificada es necesario analizar el contenido (payload) del TS que contiene la Cabecera de Cuadro de Vídeo (en hexadecimal: 00 00 01 E0).

Como se ha explicado anteriormente, el programa es capaz de estimar las pérdidas y, conociendo el tipo de imagen, estimar las pérdidas por tipo de imagen. No obstante, es importante recalcar la dificultad existente en el caso de pérdidas de cabeceras, puesto que, al perderse la información del primer paquete, se hace imposible conocer el tipo de imagen al que pertenecen los paquetes siguientes. Para estos casos, se utiliza la información de la secuencia del GOP (como una máquina de estados), aunque, debido a la flexibilidad de cambio del GOP en

el estándar H.264, también se utiliza el conocimiento del “peso” de los paquetes, es decir, del número de paquetes (RTP o TS) que componen dicha imagen “desconocida”, sabiendo que las imágenes tipo I son codificadas con mayor información y, por lo tanto, con más bytes, que las imágenes tipo P y, éstas, que las imágenes tipo B.

El programa descrito en este punto, tiene como salida la generación de un archivo de texto plano, indicando de forma periódica, las estadísticas de pérdidas a nivel de paquete (PLR) y a nivel de imagen (I, P, B). La estructura de los resultados se puede ver en la tabla 4.2.

Tabla 4.2. Estadísticas de Calidad de Servicio: pérdidas

| <i>Estadísticas QoS (PLR + IPB)</i> | |
|-------------------------------------|--|
| <i>Frame</i> | Indicador de la última imagen de la secuencia con la que se han calculado estadísticas |
| <i>PLR</i> | <i>Packet Loss Rate</i> – Tasa de pérdida de paquetes RTP |
| <i>IL</i> | <i>I Loss</i> – Porcentaje de pérdidas referidas a cuadros I |
| <i>BL</i> | <i>P Loss</i> – Porcentaje de pérdidas referidas a cuadros P |
| <i>PL</i> | <i>B Loss</i> – Porcentaje de pérdidas referidas a cuadros B |

4.5 Cálculo de la medida de Calidad de Experiencia (QoE)

El último paso para terminar la construcción de la herramienta es agregar las fórmulas de estimación de Calidad de Experiencia (obtenidas tras el ajuste del modelo de correlación) a las estadísticas de pérdida de paquetes proporcionadas por el módulo anterior.

De esta manera, consiste en aplicar las fórmulas anteriores para los casos concretos de pérdidas, obteniendo así valores estimados del MOS del vídeo.

El resultado final es un archivo de texto plano con la información de la Calidad de Servicio (QoS) con las pérdidas de paquetes, junto con la estimación de la Calidad de Experiencia (QoE) con el MOS estimado. Dicho fichero se actualiza cada 10 segundos de vídeo, tras detectar el inicio de una imagen tipo I, de tal manera que las estadísticas contienen siempre un GOP completo.

El resultado completo de la herramienta será, por lo tanto, un fichero de texto con el formato que se muestra en la figura 4.24.

```

----- Frame 251 -----
-----ESTADISTICAS QoS-----
MOS(Simple): 3.605031 - PLR: 0.364956 %
MOS (IBP): 3.802153 - IL: 0% PL: 0.00284% BL: 3.8977%
-----

```

Figura 4.24. Formato de salida herramienta monitorización

La estructura de los datos mostrados en el fichero de salida se explica en la tabla 4.3.

Tabla 4.3. Estadísticas de Calidad de Servicio (QoS) frente a Estadísticas de Calidad de Experiencia (QoE) estimadas

| <i>Estadísticas QoS vs QoE</i> | |
|--------------------------------|---|
| <i>Frame</i> | Indicador de la última imagen de la secuencia con la que se han calculado estadísticas |
| <i>MOS (Simple)</i> | <i>MOS (Mean Opinion Score)</i> – Medida subjetiva estimada de la Calidad de Experiencia, teniendo en cuenta únicamente el parámetro de Tasa de Pérdida de paquetes (PLR) |
| <i>PLR</i> | <i>Packet Loss Rate</i> – Tasa de pérdida de paquetes RTP |
| <i>MOS (IPB)</i> | <i>MOS</i> – Medida de Calidad subjetiva estimada de la Calidad de Experiencia, teniendo en cuenta los parámetros de tasas de pérdida <i>IL</i> , <i>PL</i> , <i>BL</i> |
| <i>IL</i> | <i>I Loss</i> – Porcentaje de pérdidas referidas a cuadros I |
| <i>BL</i> | <i>P Loss</i> – Porcentaje de pérdidas referidas a cuadros P |
| <i>PL</i> | <i>B Loss</i> – Porcentaje de pérdidas referidas a cuadros B |

4.6 Conclusiones

En este capítulo se ha explicado el desarrollo de la herramienta de monitorización del servicio IPTV para alta definición, detallando el funcionamiento y el resultado de cada uno de los módulos, así como la relación existente entre ellos.

Resulta de vital importancia, con el fin de conseguir unos resultados fiables, generar una muestra lo suficientemente significativa, por lo que, como en muchos otros proyectos similares, la construcción de la base de datos de vídeo y las muestras de entrenamiento son un punto clave del sistema. Se ha tratado de conseguir unas muestras heterogéneas, capaces de representar los diferentes escenarios que se pueden dar en la realidad (diferentes secuencias de vídeo con diferentes tasas de pérdidas).

Asimismo, el ajuste del modelo a los datos limita la precisión de los resultados. Debido a ello, se realizó un análisis exhaustivo y profundo de la dispersión de los datos, realizando pruebas y ajustes con diferentes variantes, combinaciones y tecnologías (gnuplot, MATLAB, SPSS), tratando de conseguir los mejores ajustes con el menor error posible.

El módulo de medición/estimación de pérdidas (Decodificador IP y estimación de calidad objetiva QoS) tiene algunas limitaciones, puesto que, al tratarse de un medidor de pérdidas, el valor obtenido nunca puede ser exacto, debido a la incertidumbre existente en cada pérdida de paquete: no podemos saber con certeza qué había en lo que se perdió. Por ello, es necesario realizar asunciones de GOP y tamaño de imágenes (peso de las mismas), con el fin de estimar, con el menor error posible, las pérdidas a nivel de imagen I, P, B.

Será en el capítulo próximo en el que se muestren los resultados obtenidos, así como se pondrá en prueba la validez y la bondad de los métodos y algoritmos implementados.

5. Validación y resultados

5.1 Introducción

Una vez explicado todo el proceso de construcción de la herramienta, vistos los puntos clave y limitaciones de la misma, en este capítulo vamos a tratar y evaluar los resultados que ofrece, con el fin de conocer las capacidades y el rendimiento que nos puede proporcionar.

Para la evaluación de la herramienta, como en otros capítulos, se procederá de forma modular, es decir, se realizará una validación por módulos. En este caso, sólo tiene sentido validar de forma independiente el ajuste del modelo a las muestras de entrenamiento, y el cálculo de las pérdidas de paquetes, puesto que son los módulos del sistema que generan una salida que repercute en el resultado final: la fórmula de cálculo de QoE y el valor de QoS.

Asimismo, como parte final de este capítulo, se presentará la evaluación global de la herramienta, comparando los resultados estimados de QoE (MOS estimado) con la secuencia de vídeo, en dos escenarios diferentes: poco movimiento y mucho movimiento (secuencia de un deporte).

5.2 Evaluación del modelo de correlación

Como se ha visto en el capítulo anterior, la primera parte de la construcción de la herramienta de monitorización se basa en la generación de un modelo de correlación, mediante las muestras de entrenamiento de las secuencias de vídeo, que explican la relación entre la Calidad de Servicio (pérdida de paquetes) y la Calidad de Experiencia (MOS).

Como en todo modelo de regresión, es necesario conocer la bondad del ajuste, es decir, saber si el ajuste que ofrece la función matemática sobre el diagrama de dispersión de la muestra de entrenamiento es suficientemente bueno. De esta manera, probamos la consistencia del modelo. Para ello, la medida más utilizada es el coeficiente de determinación, denominado R^2 . Este parámetro toma valores entre 0 y 1, siendo 0 cuando el modelo no explica la varianza de la variable aleatoria en estudio, y 1 cuando el modelo explica el 100% de la varianza.

Conjuntamente, se toman medidas de error, entre la estimación y la variable. En nuestro caso, calculamos el SEE (Error Estándar de la Estimación), representados en % en la escala MOS (1-5).

Como se explicó en el capítulo 4, la herramienta está desarrollada para dos casuísticas: IPTV con encapsulación RTP Nativo (RTP/H.264) y con MPEG-TS (RTP/MPEG-TS/H.264). Por ello, los resultados mostrados a continuación diferencian ambos casos.

La metodología seguida en las pruebas consta de un entorno de entrenamiento, formado por 20 secuencias de vídeo, y un entorno de validación, con el objetivo de validar la regresión del modelo, formado por 5 secuencias de vídeo, todas ellas de alta definición.

5.2.1 Ajuste sobre PLR

Los resultados obtenidos para el ajuste simple de las muestras de entrenamiento, considerando únicamente la tasa de pérdida de paquetes (PLR) se analizan a continuación.

Tabla 5.1. Muestras de entrenamiento RTP (PLR Simple)

| <i>Fórmula ajuste</i> | <i>R²</i> | <i>SEE (%)</i> |
|--|----------------------|-----------------------|
| $11 \times \exp(-\exp(PLR))$ | 0.73868 | 11.7519 |
| $10.9476 \times \exp(-\exp(PLR))$ | 0.73872 | 11.6613 |
| $1 + \frac{PLR}{(\frac{PLR}{2} + 1)^2}$ | 0.8239 | 10.833 |
| $1 + \frac{3.9398}{(\frac{PLR}{1.7488} + 1.0055)^2}$ | <u>0.8307</u> | 10.805 |
| $1 + \frac{3.9652}{(\frac{PLR}{2} + 1)^3}$ | 0.808 | <u>10.1587</u> |
| $1 + \frac{3.9652}{(\frac{PLR}{2.9307} + 1.0086)^3}$ | 0.82995 | 10.715 |

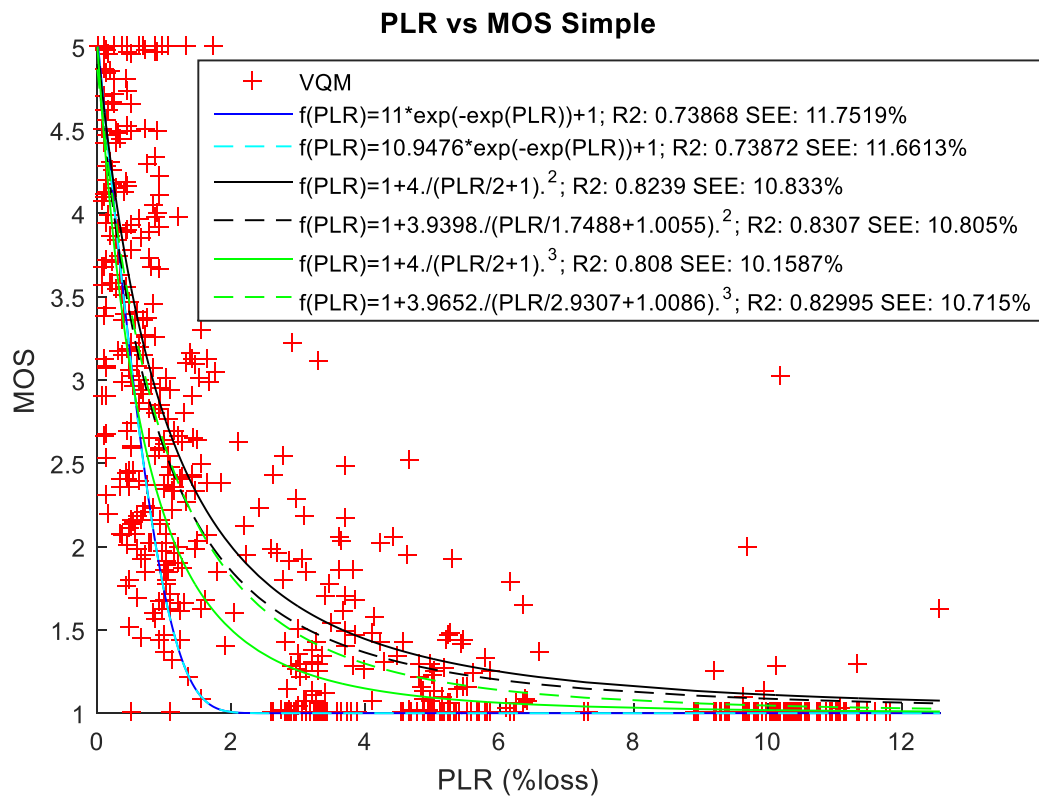


Figura 5.1 Modelo PLR Simple (Entrenamiento), RTP

Tabla 5.2. Muestras validación RTP (PLR Simple)

| <i>Fórmula ajuste</i> | <i>R²</i> | <i>SEE (%)</i> |
|--|----------------------|----------------|
| $11 \times \exp(-\exp(PLR))$ | 0.60563 | 12.3782 |
| $10.9476 \times \exp(-\exp(PLR))$ | 0.60974 | 12.4924 |
| $1 + \frac{3.9398}{(\frac{PLR}{2} + 1)^2}$ | 0.58824 | 14.0807 |
| $1 + \frac{3.9398}{(\frac{PLR}{1.7488} + 1.0055)^2}$ | <u>0.69063</u> | 11.7316 |
| $1 + \frac{3.9652}{(\frac{PLR}{2} + 1)^3}$ | 0.68447 | <u>11.5357</u> |
| $1 + \frac{3.9652}{(\frac{PLR}{2.9307} + 1.0086)^3}$ | 0.68643 | 11.6523 |

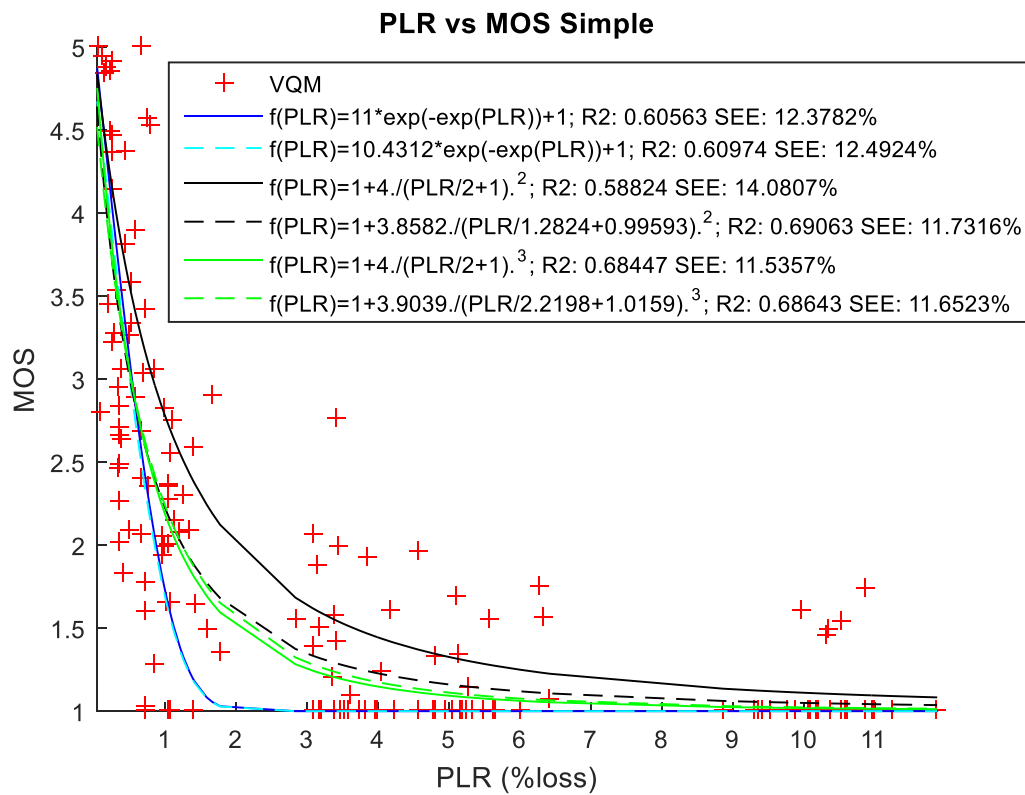


Figura 5.2. Modelo PLR Simple (Validación), RTP

Como se observa en las tablas 5.1 y 5.2, tanto para las muestras de entrenamiento como de validación las fórmulas que presentan mejores resultados son las mismas (hiperbólica cuadrada y la hiperbólica cúbica). La fórmula hiperbólica cuadrada presenta un coeficiente de determinación superior, mientras que la fórmula hiperbólica cúbica un error más bajo. En ambos casos, los dos parámetros son muy similares. No obstante, de cara a elegir el modelo que mejor explica la distribución de los datos, elegimos la fórmula hiperbólica cuadrada (5.1), puesto que es aquella que presenta un mejor coeficiente de determinación y, aunque no es la que tiene menor error, se aproxima bastante al error mínimo.

$$1 + \frac{3.9398}{(\frac{PLR}{1.7488} + 1.0055)^2} \quad (5.1)$$

Tabla 5.3. Muestras entrenamiento MPEG-TS (PLR Simple)

| Fórmula ajuste | R^2 | SEE (%) |
|--|-----------------------|---------------------|
| $11 \times \exp(-\exp(PLR))$ | 0.83149 | 9.7034 |
| $10.8267 \times \exp(-\exp(PLR))$ | 0.83198 | 9.5594 |
| $1 + \frac{4}{(\frac{PLR}{2} + 1)^2}$ | 0.83667 | 10.5488 |
| $1 + \frac{3.959}{(\frac{PLR}{1.3384} + 0.99803)^2}$ | <u>0.87164</u> | 9.192 |
| $1 + \frac{4}{(\frac{PLR}{2} + 1)^3}$ | 0.86973 | <u>8.816</u> |
| $1 + \frac{3.984}{(\frac{PLR}{2.22} + 1.0029)^3}$ | 0.8715 | 9.0712 |

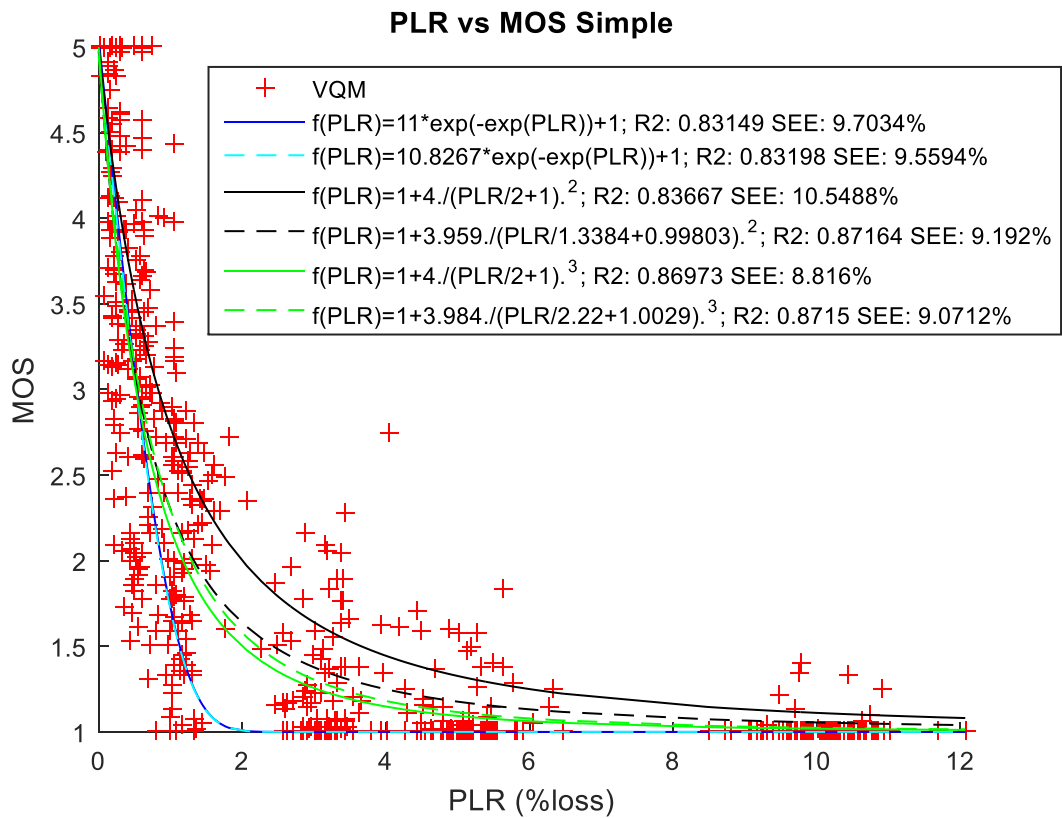


Figura 5.3 Modelo PLR Simple (Entrenamiento), MPEG-TS

Tabla 5.4. Muestras validación MPEG-TS (PLR Simple)

| Fórmula ajuste | R^2 | SEE (%) |
|---------------------------------------|----------------|----------------|
| $11 \times \exp(-\exp(PLR))$ | 0.5646 | 13.2545 |
| $10.8267 \times \exp(-\exp(PLR))$ | 0.58017 | 13.3107 |
| $1 + \frac{4}{(\frac{PLR}{2} + 1)^2}$ | 0.50878 | 14.6949 |

| | | |
|---|----------------|----------------|
| $1 + \frac{3.959}{\left(\frac{PLR}{1.3384} + 0.99803\right)^2}$ | 0.64504 | 13.29 |
| $1 + \frac{4}{\left(\frac{PLR}{2} + 1\right)^3}$ | 0.63108 | 13.0355 |
| $1 + \frac{3.984}{\left(\frac{PLR}{2.22} + 1.0029\right)^3}$ | 0.64165 | 13.1857 |

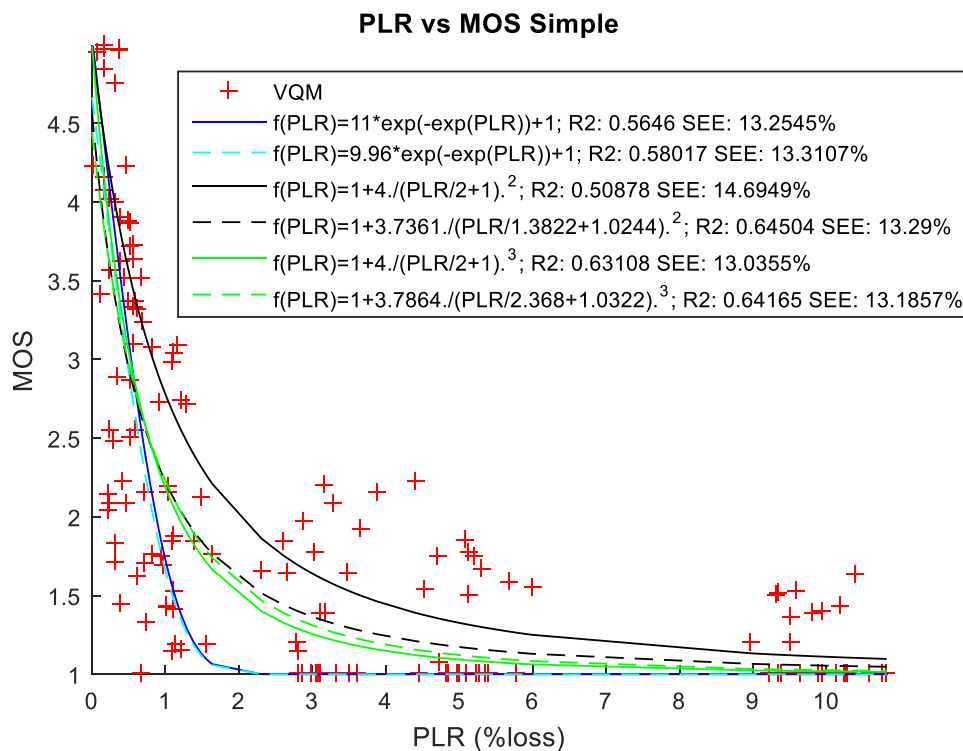


Figura 5.4. Modelo PLR Simple (Validación), MPEG-TS

Como se observa en las tablas 5.3 y 5.4, caso similar para RTP, las fórmulas que presentan mejores resultados son las mismas (hiperbólica cuadrada y la hiperbólica cúbica). Por la misma razón que en el caso RTP, elegimos la fórmula hiperbólica cuadrada (5.2), puesto que es aquella que presenta un mejor coeficiente de determinación y, aunque no es la que tiene menor error, se aproxima bastante al error mínimo.

$$1 + \frac{3.959}{\left(\frac{PLR}{1.3384} + 0.99803\right)^2} \quad (5.2)$$

Tabla 5.5 Comparativa RTP vs MPEG-TS (PLR Simple)

| | <i>Fórmula ajuste</i> | <i>R²</i> | <i>SEE (%)</i> |
|----------------|---|----------------------|----------------|
| <i>RTP</i> | $1 + \frac{3.9398}{\left(\frac{PLR}{1.7488} + 1.0055\right)^2}$ | 0.8307 | 10.805 |
| <i>MPEG-TS</i> | $1 + \frac{3.959}{\left(\frac{PLR}{1.3384} + 0.99803\right)^2}$ | 0.87164 | 9.192 |

Como se observa en la tabla 5.5, para ambas encapsulaciones de IPTV, la fórmula elegida es similar (aunque con diferentes valores): función hiperbólica cuadrada. De la comparación RTP frente a MPEG-TS también sacamos la conclusión de que el modelo MPEG-TS consigue mejores resultados de ajuste (0.87164 frente a 0.8307 de RTP). La explicación de este hecho puede ser la no existencia de muestras dispersas (que sí existen en el caso de RTP Nativo), correspondientes a vídeos con altas pérdidas pero que mantienen una calidad razonable de visionado. Esto viene determinado por lo explicado en el apartado 2.2.3.2, en el que se explicaban las ventajas de la encapsulación RTP Nativo para la transmisión de vídeo H.264, frente a MPEG-TS.

Por último, vemos que los resultados de error están entorno al 9-10%, resultados bastante positivos, teniendo en cuenta que el algoritmo no tiene conocimiento ni referencia alguna del vídeo original.

5.2.2 Ajuste sobre IPB

Los resultados obtenidos para el ajuste IPB de las muestras de entrenamiento, considerando las tasas de pérdidas de imágenes I (IL , $I Loss$), de imágenes P (PL , $P Loss$), y de imágenes B (BL , $B Loss$), se muestran a continuación.

Tabla 5.6. Muestras entrenamiento RTP (IPB)

| <i>Fórmula ajuste</i> | R^2 | SEE (%) |
|--|----------------|----------------|
| <i>Combinación función exponencial (IL, PL, BL)</i> | 0.82403 | 12.7457 |
| <i>Combinación función hiperbólica cuadrada (IL, PL, BL)</i> | 0.86109 | 9.6531 |
| <i>Combinación función hiperbólica cúbica (IL, PL, BL)</i> | 0.85801 | 9.6861 |
| <i>Combinación función con mejor ajuste independiente (IL, PL, BL)</i> | 0.85586 | 10.4639 |

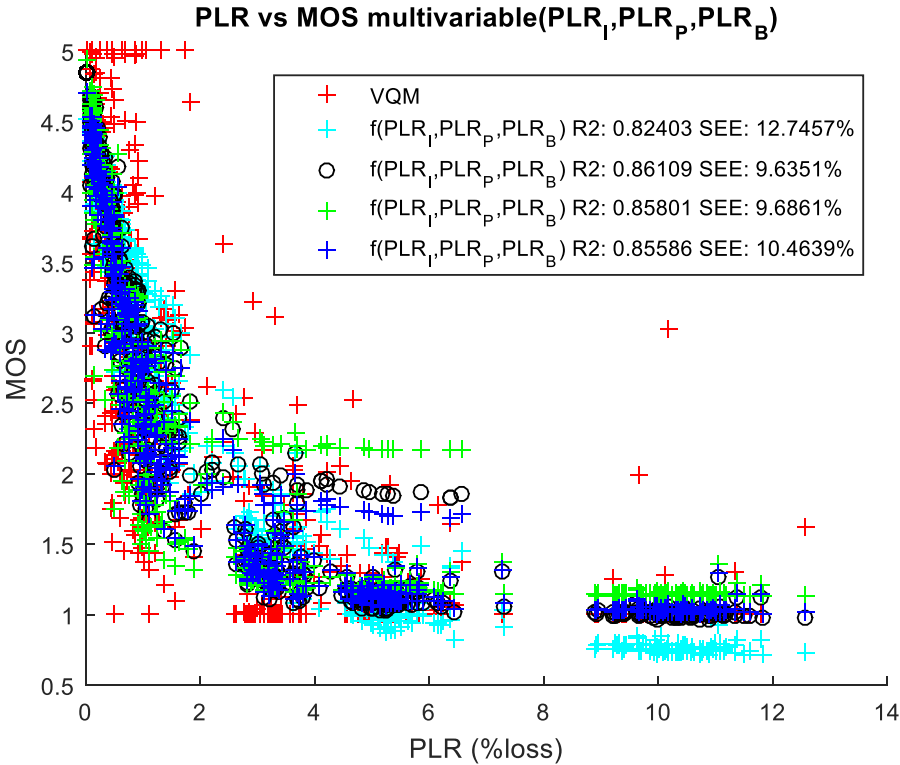


Figura 5.5 Modelo PLR Simple (Entrenamiento), RTP

Tabla 5.7. Muestras validación RTP (IPB)

| <i>Fórmula ajuste</i> | <i>R²</i> | <i>SEE (%)</i> |
|---|----------------------|----------------|
| <i>Combinación función exponencial (IL, PL, BL)</i> | 0.67013 | 12.6913 |
| <i>Combinación función hiperbólica cuadrada (IL, PL, BL)</i> | 0.71963 | 11.506 |
| <i>Combinación función hiperbólica cúbica (IL, PL, BL)</i> | 0.71877 | 11.6955 |
| <i>Combinación función con mejor ajuste independiente (IL, PL, BL)</i> | 0.71707 | 11.6783 |

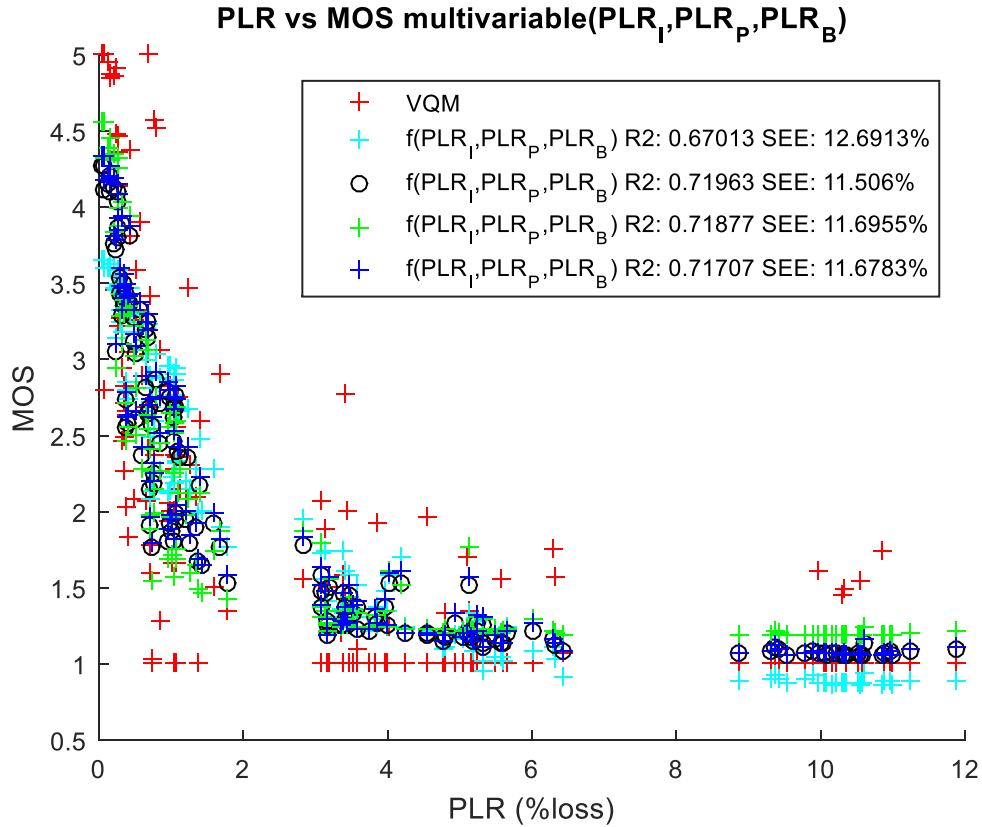


Figura 5.6. Modelo IPB Simple (Validación), RTP

Como se observa en las tablas 5.6 y 5.7, tanto para las muestras de entrenamiento como de validación, la fórmula que proporciona una mejor fiabilidad de estimación de la Calidad Percibida (MOS) es la formada por la combinación de las funciones hiperbólicas cuadradas (5.3). Resulta curioso que la fórmula formada por la combinación de las funciones con mejor ajuste de forma independiente en IL, BL y PL no sea la que ofrece los mejores resultados de forma conjunta. Este hecho demuestra que, aunque las pérdidas de cada tipo de imagen por separado se ajusten mejor a un determinado modelo, prevalece la importancia del modelo conjunto frente a cada modelo independiente.

$$\begin{aligned}
 MOS = & -0.027 + 0.20 \times f_{\text{hiperbólica cuadrada(IL)}} + 0.527 \times f_{\text{hiperbólica cuadrada(PL)}} \\
 & + 0.247 f_{\text{hiperbólica cuadrada(BL)}}
 \end{aligned}
 \tag{5.3}$$

Siendo:

$$f_{hiperbólica_{cuadrada}(IL)} = f_{hiperbólica_{cuadrada}(PL)} = f_{hiperbólica_{cuadrada}(BL)} = 1 + \frac{4}{\left(\frac{PLR}{2} + 1\right)^2}$$

Tabla 5.8. Muestras entrenamiento MPEG-TS (IPB)

| Fórmula ajuste | R ² | SEE (%) |
|--|----------------|----------------|
| Combinación función exponencial (IL, PL, BL) | 0.8322 | 12.9177 |
| Combinación función hiperbólica cuadrada (IL, PL, BL) | 0.88951 | 9.3043 |
| Combinación función hiperbólica cúbica (IL, PL, BL) | 0.899 | 8.2995 |
| Combinación función con mejor ajuste independiente (IL, PL, BL) | 0.87653 | 10.4775 |

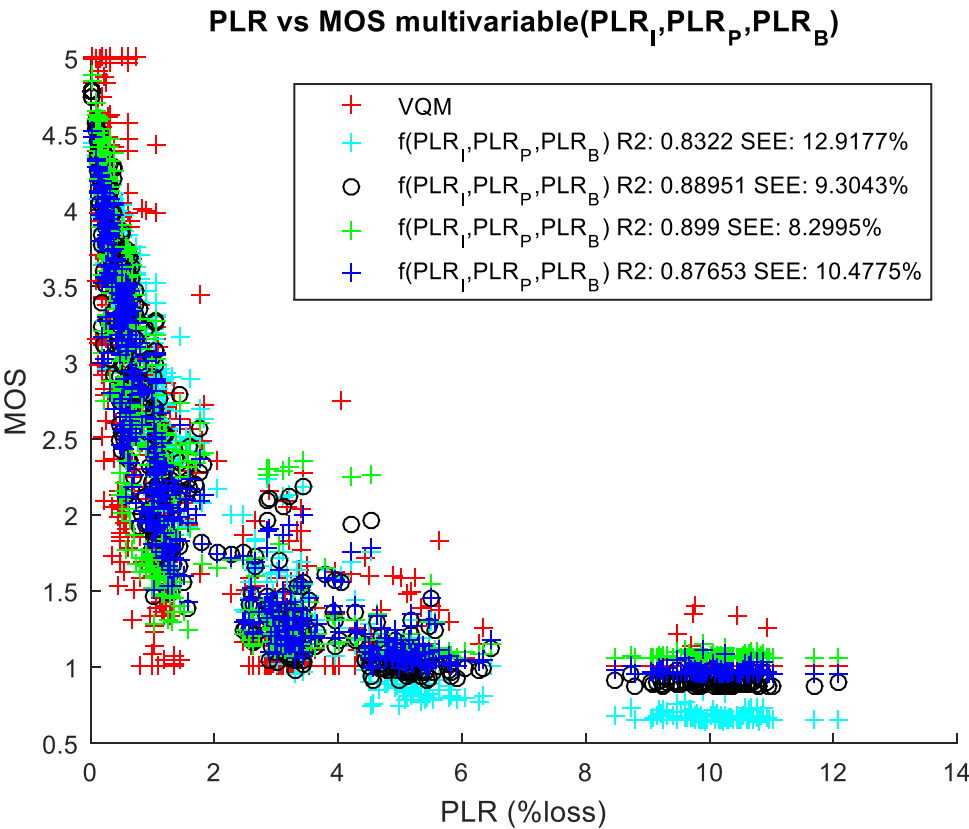


Figura 5.7 Modelo IPB (Entrenamiento), MPEG-TS

Tabla 5.9. Muestras validación MPEG-TS (IPB)

| Fórmula ajuste | R ² | SEE (%) |
|--|----------------|----------------|
| Combinación función exponencial (IL, PL, BL) | 0.60447 | 13.9514 |
| Combinación función hiperbólica cuadrada (IL, PL, BL) | 0.62582 | 13.868 |
| Combinación función hiperbólica cúbica (IL, PL, BL) | 0.61899 | 14.0615 |
| Combinación función con mejor ajuste independiente (IL, PL, BL) | 0.62433 | 13.8065 |

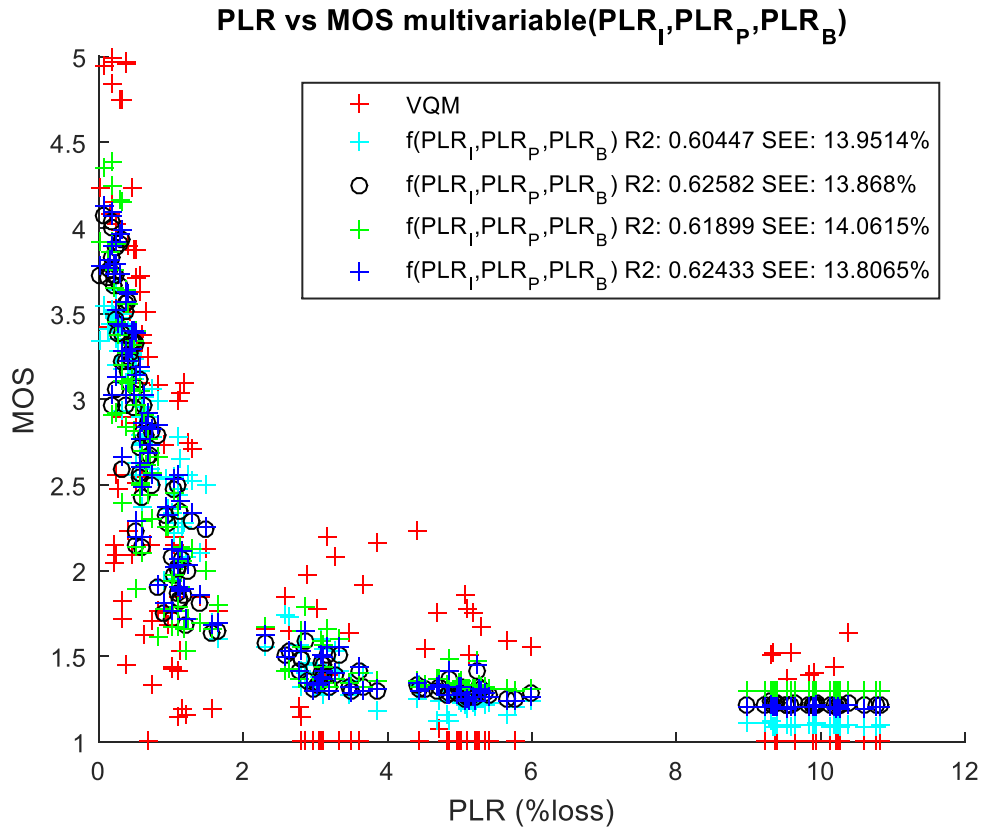


Figura 5.8. Modelo IPB (Validación), MPEG-TS

En el caso MPEG-TS, como se observa en las tablas 5.8 y 5.9, los resultados de entrenamiento y validación no arrojan una evidencia clara. Por un lado, en los datos de entrenamiento se obtiene como mejor fórmula la función hiperbólica cúbica (tanto por coeficiente de determinación como por mínimo error. Por otro lado, en los datos de validación se obtiene como mejores fórmulas la función hiperbólica cuadrada (mejor coeficiente de determinación y un error similar al error mínimo) y la función combinada (mínimo error). No obstante, la función hiperbólica cúbica presenta valores cercanos a los máximos, tanto en el coeficiente de determinación y en el error. Por todo ello, se elige como mejor fórmula de ajuste a la función hiperbólica cúbica (5.4).

$$\begin{aligned}
 MOS = & -0.1348 + 0.2515 \times f_{hiperbólica\acute{c}ubica(IL)} + 0.5079 \times f_{hiperbólica\acute{c}ubica(PL)} \\
 & + 0.2237 f_{hiperbólica\acute{c}ubica(BL)}
 \end{aligned}
 \tag{5.4}$$

Siendo:

$$f_{hiperbólica\acute{c}ubica(IL)} = f_{hiperbólica\acute{c}ubica(PL)} = f_{hiperbólica\acute{c}ubica(BL)} = 1 + \frac{4}{\left(\frac{PLR}{2} + 1\right)^3}$$

Tabla 5.10 Comparativa RTP vs MPEG-TS (IPB)

| | <i>Fórmula ajuste</i> | <i>R²</i> | <i>SEE (%)</i> |
|----------------|--|----------------------|----------------|
| <i>RTP</i> | Combinación función hiperbólica cuadrada (IL, PL, BL) | 0.86109 | 9.6531 |
| <i>MPEG-TS</i> | Combinación función hiperbólica cúbica (IL, PL, BL) | 0.899 | 8.2995 |

Como ocurría en el ajuste sobre PLR Simple, los resultados de la tabla 5.10 muestran que el modelo MPEG-TS presenta un mayor ajuste que el modelo RTP (0.88951 frente a 0.86109). En ambos casos, los resultados de error están por debajo del 10%, resultados muy positivos para el algoritmo implementado.

Tabla 5.11. Comparativa PLR vs IPB

| | <i>Fórmula ajuste</i> | <i>R²</i> | <i>SEE (%)</i> |
|----------------------|---|----------------------|----------------|
| <i>RTP (PLR)</i> | $1 + \frac{3.9398}{\left(\frac{PLR}{1.7488} + 1.0055\right)^2}$ | 0.8307 | 10.805 |
| <i>RTP (IPB)</i> | Combinación función hiperbólica cuadrada (IL, PL, BL) | 0.86109 | 9.6531 |
| <i>MPEG-TS (PLR)</i> | $1 + \frac{3.959}{\left(\frac{PLR}{1.3384} + 0.99803\right)^2}$ | 0.87164 | 9.192 |
| <i>MPEG-TS (IPB)</i> | Combinación función hiperbólica cúbica (IL, PL, BL) | 0.899 | 8.2995 |

En la tabla 5.11 se muestra la comparativa de los resultados del ajuste de ambos modelos: PLR Simple frente a IPB. El principal hecho destacable que reflejan los resultados es que el modelo IPB es ligeramente superior frente al modelo PLR, algo que era de esperar, puesto que se trata de un modelo más complejo, con mayor información de la secuencia de vídeo. Se obtiene un ajuste de 0,899 frente a 0,86, para MPEG-TS, y un ajuste de 0.87 frente a 0.83 en RTP. Asimismo, los valores de error pasan de 9,6-10,8 a 8,3-9,2, una diferencia de un 1,5% mejor en IPB.

Por otro lado, también observamos algo que hemos ido comentando en páginas anteriores, en la encapsulación MPEG-TS se obtiene un modelo más ajustado que para el caso de RTP Nativo (como se explicó en el capítulo 4.3).

5.3 Evaluación del decodificador de paquetes IP

El módulo de decodificación de paquetes IP es el encargado de analizar las capturas *pcap* de las secuencias de vídeo IPTV, con el fin de proporcionar una medición de pérdidas de paquetes. Como se explicó en el capítulo anterior, contar las pérdidas de paquetes siempre conlleva un error, puesto que se trata de una estimación de cuántos paquetes se han perdido y qué tipo de imagen (I, P, B) codificaban.

Por ello, en la tabla 5.12 se muestra el resultado de la comparación de las pérdidas reales (a través de la referencia del *pcap* original) frente a las pérdidas calculadas por el módulo de la herramienta (únicamente *pcap* degradado).

Tabla 5.12 Error medición tasas de pérdidas

| | <i>Error (%)</i> |
|------------|------------------|
| <i>PLR</i> | 1.8 |
| <i>IL</i> | 1.7 |
| <i>PL</i> | 1.67 |
| <i>BL</i> | 2.27 |

En las figuras 5.9, 5.10, 5.11 y 5.12 se observan los histogramas de los errores de tasa de pérdidas. En ellas se observa que no existen grandes errores, siendo 5.5% el máximo (para el caso de BL).

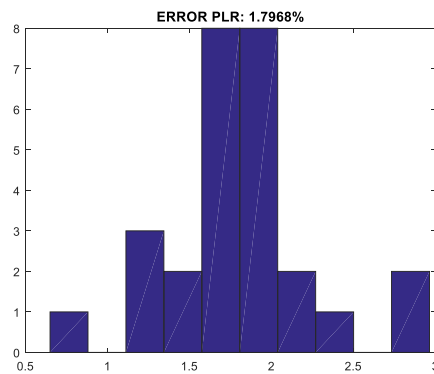


Figura 5.9. Histograma error tasa PLR

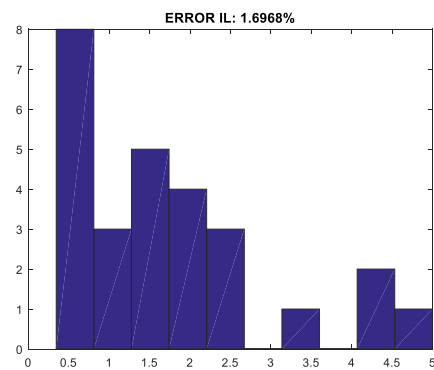


Figura 5.10 Histograma error tasa IL

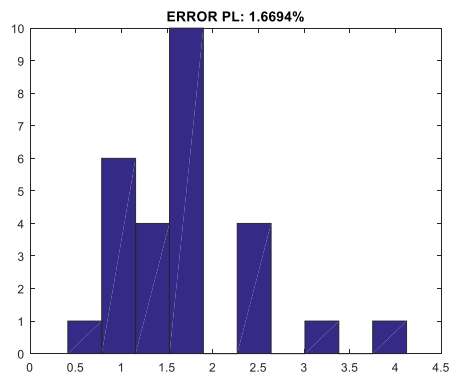


Figura 5.11 Histograma error tasa PL

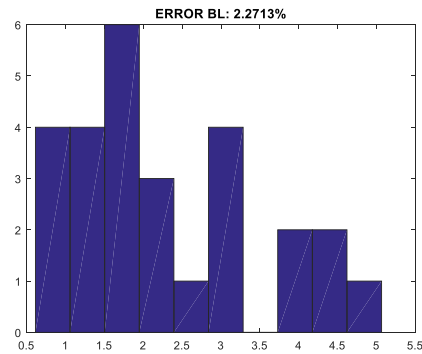


Figura 5.12. Histograma error tasa BL

5.4 Evaluación global del sistema

Una vez evaluados los resultados de forma modular (puntos 5.2 y 5.3), vamos a mostrar los resultados de la herramienta de forma completa, es decir, se va a realizar el análisis de dos secuencias de vídeo, mostrando los resultados de Calidad de Experiencia devueltos por la propia herramienta.

Dado que la QoE es una medida subjetiva, la evaluación global del sistema seguirá una valoración subjetiva de la imagen, comparando los resultados de MOS estimado, con la degradación que se puede observar en los píxeles de la imagen, en determinados momentos de la secuencia de vídeo.

Con el fin de evaluar la potencia de la herramienta, se han seleccionado dos secuencias de vídeo con escenas totalmente diferentes: secuencia con poco movimiento y secuencia con mucho movimiento.

5.4.1 Secuencia con poco movimiento

Se trata de una secuencia de vídeo en la que hay algo de movimiento de las personas que aparecen en escena, pero sin llegar a realizarse muchos cambios de cámara ni producirse movimientos rápidos de ningún elemento de la escena.

La herramienta de monitorización devuelve las siguientes estadísticas:

```

----- Frame 211 -----
-----ESTADISTICAS QoS-----
MOS(Simple): 5.000000 - PLR: 0.000000 %
MOS(IPB): 5.000000 - IL: 0.000000 % PL: 0.000000 % BL: 0.000000 %
-----

```

Figura 5.13. Secuencia con poco movimiento, intervalo 113-211



Figura 5.14. Secuencia con poco movimiento, Frame 209

Frame 211: en este intervalo, las estadísticas calculan una tasa de pérdidas del 0%, por lo que el MOS estimado es de 5, tanto para el modelo Simple (PLR) y el modelo IPB.

```

----- Frame 606 -----
-----ESTADISTICAS QoS-----
MOS(Simple): 4.226106 - PLR: 0.146889 %
MOS(IPB): 3.966707 - IL: 1.473684 % PL: 0.000000 % BL: 0.000000 %
-----

```

Figura 5.15. Secuencia con poco movimiento, intervalo 501-606



Figura 5.16. Secuencia con poco movimiento, Frame 557

Frame 606: en este intervalo, se observa una degradación en la parte inferior de la imagen. Se trata de una degradación que se mantiene durante todo el intervalo, causado por la pérdida de información de una imagen de referencia (imagen I). Como vemos en las estadísticas, tenemos una tasa de pérdidas de imagen I de 1.47%, lo que explica la causa de dicha degradación. El MOS estimado en el modelo IPB es de 3.99, mientras que en el modelo PLR (Simple) es algo mayor: 4.22. Vista la imagen (figura 5.16) parece que el resultado que más se acerca es el modelo IPB, puesto que estamos ante una situación en la que la tasa de pérdidas no es alta (0.15%), pero sí es significativa la tasa de pérdidas de imagen I (1.47%). En estos casos, el modelo Simple, al no tener en cuenta el tipo de imagen, ofrece una estimación menos fiable que el modelo IPB.

```

----- Frame 788 -----
-----ESTADISTICAS QoS-----
MOS(Simple): 3.713431 - PLR: 0.280899 %
MOS(IPB): 4.239234 - IL: 0.000000 % PL: 0.000000 % BL: 0.726111 %
-----

```

Figura 5.17. Secuencia con poco movimiento, intervalo 690-788



Figura 5.18. Secuencia poco movimiento, Frame 780

Frame 788: en este intervalo, la secuencia de vídeo prácticamente no presenta ninguna degradación visible al ojo humano (figura 5.18). En las estadísticas observamos una tasa de pérdidas totales de 0.3%, una tasa de pérdidas que ya podría implicar una degradación de la imagen (por ello el MOS estimado Simple indica degradación). No obstante, vemos que las pérdidas existentes corresponden a imágenes de tipo B, las cuales no tienen gran influencia en la degradación del vídeo (por ello, el MOS estimado IPB ofrece un resultado más acorde con la realidad).

5.4.2 Secuencia con mucho movimiento

En este segundo escenario, estamos ante una secuencia de vídeo con alto grado de movimiento, puesto que se trata de una secuencia de vídeo de un partido de baloncesto. En dicha secuencia existen múltiples componentes realizando movimientos, así como cambios de cámara y de velocidad de reproducción (repeticiones a cámara lenta).

La herramienta de monitorización devuelve las siguientes estadísticas:

```
----- Frame 251 -----
-----ESTADISTICAS QoS-----
MOS(Simple): 3.451840 - PLR: 0.364956 %
MOS(IPB): 3.623946 - IL: 0.000000 % PL: 0.463196 % BL: 0.189343 %
-----
```

Figura 5.19. Secuencia con mucho movimiento, intervalo 1-251



Figura 5.20. Secuencia con mucho movimiento, frame 69

Frame 251: en este intervalo de tiempo, la secuencia de vídeo presenta una degradación en la parte inferior de la imagen (figura 5.20), principalmente en lo relativo al marcador y a las zonas de más movimiento (jugadores en carrera). El MOS estimado por ambos métodos es muy similar, indicando una visualización molesta de la imagen.

```
----- Frame 501 -----
-----ESTADISTICAS QoS-----
MOS(Simple): 4.386469 - PLR: 0.111359 %
MOS(IPB): 4.293679 - IL: 0.000000 % PL: 0.174467 % BL: 0.027718 %
-----
```

Figura 5.21. Secuencia con mucho movimiento, intervalo 251-501

Frame 501: en este segundo intervalo de tiempo, las estadísticas proporcionadas por ambos métodos indican una visualización más o menos correcta, con alguna imperfección, pero no molesta. Sin embargo, como podemos ver en la figura 5.22, existe una degradación importante en muchas partes de la imagen, siendo molesta para la visualización. En este caso,

el MOS debería haber sido inferior a la estadística proporcionada por ambas implementaciones, si bien, el método más complejo identifica mejor la degradación (figura 5.21).



Figura 5.22 Secuencia con mucho movimiento, frame 371

Como se ha podido ver, mediante un análisis visual y subjetivo de las secuencias de vídeo, las estadísticas proporcionadas por la herramienta de monitorización cumplen con su objetivo, siendo consistentes y detectando degradaciones y visualizaciones molestas. También se ha podido observar la clara ventaja que ofrece el análisis en detalle del tipo de imagen, como el caso del modelo IPB, evitando fallos en casos de altas tasas de pérdidas, pero con poca degradación (como en el intervalo 788 de la secuencia con poco movimiento); y casos de bajas pérdidas, pero con alta degradación en la imagen (como en el intervalo 606 de la secuencia con poco movimiento).

5.5 Conclusiones

En este capítulo se ha realizado un análisis en detalle de los modelos utilizados para estimar la herramienta, en los cuales conseguimos siempre un ajuste superior al 0.75, considerándose, así, como buenas estimaciones.

Por otro lado, el módulo encargado de realizar las mediciones de pérdidas ofrece unas tasas de error bajas. Teniendo en cuenta las limitaciones existentes en este tema, debido a la incertidumbre que presenta el hecho de recuperar la información perdida, se pueden considerar como buenos para el propósito de la herramienta.

Asimismo, hemos mostrado el comportamiento de nuestra herramienta ante dos escenarios reales. Se puede considerar que la respuesta de la misma es la esperada, exceptuando casos en los que existe alto grado de movimiento, puesto que somos capaces de estimar una Calidad de Experiencia acorde de lo esperado tras un análisis visual de la imagen.

Un aspecto importante a destacar de la herramienta es que el objetivo de la misma no es determinar con exactitud el valor del MOS estimado (5; 4,5 o 4), puesto que ni el propio ser humano sería capaz de diferenciar si el vídeo que está visualizando tiene una calidad en la escala MOS de 4 o 4,5. La ventaja y el objetivo de esta herramienta de monitorización es de informar al Proveedor de Servicios de Internet, el encargado de suministrar el servicio de IPTV al cliente y ofrecerle una calidad adecuada, los momentos en los que el usuario está teniendo una satisfacción baja en la visualización del vídeo. Es decir, la herramienta es capaz de detectar y seguir tendencias, tanto positivas como negativas, de Calidad de Experiencia.

6. Conclusiones y Trabajos Futuros

6.1 Resumen del trabajo

El objetivo de este Trabajo Final de Máster ha sido implementar y desarrollar una herramienta de monitorización de vídeo IPTV de alta definición, capaz de estimar la Calidad de Experiencia del usuario, mediante parámetros de calidad de la red: tasa de pérdida de paquetes.

Como se ha explicado en capítulos anteriores, existen diferentes alternativas para medir la QoE, siendo los métodos sin referencia los que mejor tiempo de respuesta ofrecen y menores requisitos presentan para su implantación (al no precisar de ningún tipo de información de la señal original). En este trabajo se ha desarrollado una herramienta de medida de calidad percibida mediante un método indirecto. Estos métodos precisan de la generación de un modelado matemático que correlaciona la información de calidad de servicio y la calidad de experiencia.

6.2 Conclusiones

Para conseguir los objetivos propuestos y la elaboración de este proyecto precisamos de secuencias de vídeos de alta definición (1920x1080 píxeles). Estos vídeos presentan tamaños elevados (aproximadamente 1GB para una duración de 15 segundos de vídeo sin codificar). Este hecho ha generado problemas en el cálculo de diferentes parámetros, en especial del parámetro VQM, debido a las elevadas prestaciones de memoria necesitadas por el software empleado en su cálculo.

Es de destacar la importancia que tiene la calidad del modelado (bondad del ajuste) en el resultado global de la herramienta. Conseguir un modelo que explique la distribución de las variables de calidad de la mejor manera posible es un punto clave en la implementación, la cual nos delimitará la validez de los resultados finales. Por ello, en este trabajo se ha realizado un trabajo en profundidad en este campo, consiguiendo resultados muy positivos en lo referente a los coeficientes de correlación y errores de estimación.

Con el objetivo de mejorar los resultados ofrecidos por la herramienta, no realizamos un análisis simple de las pérdidas de paquetes, sino que profundizamos en un análisis dependiente del tipo de imagen (pérdidas de imagen I, imagen P e imagen B). Este análisis en detalle introduce una complejidad en el proceso, tanto a nivel de modelado como a nivel de medida de tasas de pérdidas. No obstante, los resultados obtenidos justifican esta complejidad, puesto que se mejora el ajuste del modelo de correlación y, sobretodo, se obtienen resultados de estimación de calidad de experiencia más próximos a la realidad. Aun así, existen escenarios en los que los resultados obtenidos no se asemejan con lo esperado, como en el caso de secuencias de vídeo con mucho movimiento.

En referencia al cálculo de las tasas de pérdidas, es importante mencionar las dificultades y limitaciones encontradas. A diferencia de otros estándares de codificación, H.264 presenta un GOP dinámico en función de las características de la escena, lo que provoca mayores dificultades a la hora de conocer el tipo de imagen codificada en los paquetes perdidos. A pesar de ello, los resultados obtenidos (capítulo 5) presentan tasas de error bajas, teniendo poco impacto en la estimación final de la Calidad de Experiencia (MOS).

Asimismo, nos hemos encontrado con dificultades en la búsqueda de información sobre el formato del flujo de bits (*bitstream*) en el caso de H.264 con MPEG-TS. La documentación existente está referida para el caso de vídeo de calidad estándar (MPEG-2), pero el caso de vídeo

de alta definición (H.264) presenta importantes diferencias y la ausencia de cabeceras (inicio de secuencia, inicio de GOP, etc). Debido a esto, fue necesario realizar un análisis ad-hoc del flujo de bits de vídeos de ejemplo, descubriendo los patrones existentes en el mismo.

Vistos los resultados globales de la herramienta, concluimos que la misma es capaz de detectar anomalías (calidades muy bajas) y seguir las tendencias de calidad presentes en la visualización de la red, siendo posible obtener información de si el servicio proporcionado está siendo satisfactorio o no, independientemente del valor exacto del MOS.

En definitiva, pese a las limitaciones encontradas, principalmente debidas por tratarse de vídeo de alta definición codificado en H.264, se han cumplido los objetivos propuesto en este Trabajo Final de Máster, proporcionando estadísticas de calidad de experiencia, en tiempo real, utilizando únicamente medidas de la calidad de la red: tasa de pérdida de paquetes.

6.3 Trabajos futuros

Analizados los resultados de la herramienta, conociendo las limitaciones de la misma, se proponen una serie de puntos de mejora y trabajos futuros para, manteniendo la misma línea de investigación, conseguir mejoras en los resultados de estimación de calidad percibida:

- Construir una base de datos de vídeos de alta definición más amplia, con una mayor diversidad, incluyendo vídeos con diferentes parámetros de codificación (frecuencia de imagen, frecuencia de fotogramas de refresco, tamaño de la imagen, etc).
- Incluir el análisis de otros parámetros, tales como la información de movimiento: SI (*Spatial Information*) y TI (*Temporal Information*) [23], con el fin de mejorar los resultados en escenarios específicos con mucho movimiento (ej: deportes).
- Completar las muestras de entrenamiento con una evaluación subjetiva (MOS), con la colaboración de un grupo de personas, proporcionando una mayor robustez a las muestras.
- Investigar la paralelización del algoritmo, permitiendo una ejecución simultánea, consiguiendo resultados para diferentes secuencias de vídeo en el mismo instante temporal, pudiendo cruzar información para consolidar los resultados de estimación de calidad percibida.
- Desarrollar modelos matemáticos más complejos para la estimación de la calidad, independientemente de la complejidad y el tiempo necesario para su cálculo.
- Desarrollo de una interfaz web en la que sea posible visualizar la secuencia de vídeo junto con las estadísticas devueltas por la herramienta de monitorización.

Referencias

- [1] D. Hernando, "Desarrollo de un sistema de medición, monitorización y gestión de redes IPTV", *Trabajo Final de Grado*, Universidad Autónoma de Madrid (UAM), Jun. 2013.
- [2] A. MacAulay, B. Felts and Y. Fisher, "IP Streaming of MPEG-4: Native RTP vs MPEG-2 Transport Stream", Oct. 2005.
- [3] P. de la Cruz. "Contribución a los modelos y metodologías para la estimación de la calidad percibida por los usuarios (QoE) a partir de parámetros de calidad de red/servicio (QoS) en servicios convergentes multimedia (Triple-Play)", *Tesis Doctoral*, Universidad Politécnica de Madrid (UPM), 2012.
- [4] M. Ellis and C. Perkins, "Packet Loss Characteristics of IPTV-live Traffic on Residential Links", in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, Jan, pp. 1-5.
- [5] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services", Feb. 2016.
- [6] G. Halinger and O. Hohlfeld, "The gilbert-elliott model for packet loss in real time services on the internet", *14th MMB Conference Dortmund Germany*, pp. 269-283, 2008.
- [7] S. Salsano, F. Ludovici, A. Ordine and D. Giannuzi, "Definition of a general and intuitive loss model for packet networks and its implementation in the Netem module in the Linux kernel", Technical report, University of Rome, 2009.
- [8] P. Selling and M. Reisslein, "Video Transport Evaluation With H.264 Video Traces", in *IEEE Communications Surveys & Tutorials*, volumen 14, pp. 1142-1165, 2011.
- [9] S. Paulikas, "Estimation of video quality of H.264/AVC video streaming", in *IEEE Eurocon 2013*, Oct. 2013.
- [10] K. Kim, "MPEG-2 ES/PES/TS/PSI", Kyung-Hee University, Oct. 2010.
- [11] S. Wolf, M. Pinson, "Video Quality Measurement Techinques", NTIA Technical Report TR-02-392, Jun. 2002.
- [12] Architecture & Transport Working Group, "Triple-play Services Quality of Experience (QoE) Requirements", Technical Report TR-126, Dec. 2006
- [13] I. Sedano, K. Brunnström, M. Kihl and A. Aurelius, "Full-reference video quality metric assisted the development of no-reference bitstream video quality metrics for real-time networks monitoring", in *EURASIP Journal on Image and Video Processing*, 2014.
- [14] "FFmpeg." [Online]. Available: <http://www.ffmpeg.org/>
- [15] Video Test Media, [Online]. Available: <https://media.xiph.org/video/derf/>
- [16] J.C. Cuéllar, J.H. Ortiz and J.L. Arciniegas, "Clasificación y Análisis de Métodos para medir Calidad de la Experiencia del Servicio de Televisión sobre Protocolo IP (IPTV)", in *Información Tecnológica* 25(5), pp. 121-128, Dec. 2013.
- [17] "Netem" [Online]. Available: <http://netgroup.uniroma2.it/twiki/bin/view.cgi/Main/NetemCLG>

- [18] "EOS Movie Compression" [Online]. Available:
http://www.canon.com.hk/cpx/en/technical/va_EOS_Movie_Compression_Options_All_I_and_IPB.html
- [19] ITU-T Rec. P.10/G.100, "Vocabulario sobre calidad de funcionamiento y de servicio", 2006.
- [20] L. Salgado, "Tema 5: Estándares H.263, MPEG-4, H.264/AVC", en la asignatura *Tecnologías de Vídeo*, 2013.
- [21] F. Díaz and M. de Frutos, "El estándar de codificación de vídeo H.264/AVC", Universidad Carlos III de Madrid, 2008.
- [22] M. Borella, D. Swider, S. Uludag, and G. Brewster, "Internet packet loss: measurement and implications for end-to-end qos", in *Architectural and OS Support for Multimedia Applications" Flexible Communication Systems*, Wireless Networks and Mobile Computing., 1998, *Proceedings of the 1998 ICPP Workshops* on, Aug, pp. 3-12.
- [23] O. Nemethova, M. Ries and M. Rupp, "Quality assessment for H.264 coded low-rate and low-resolution video sequences", in *International Conference on Communications, Internet and Information Technology (IASTED)*, Nov. 2004, pp. 22-24

A.1 Bitstream

A.1.1 RTP

A.1.1.1 NAL Unit Type

| NAL Unit Type | Packet Type | Packet Type Name | |
|---------------|------------------------------|--------------------------------|--|
| 0 | reserved | | |
| 1-23 | NAL unit | Single NAL unit packet | |
| 24 | STAP-A | Single-time aggregation packet | |
| 25 | STAP-B | Single-time aggregation packet | |
| 26 | MTAP16 | Multi-time aggregation packet | |
| 27 | MTAP24 | Multi-time aggregation packet | |
| 28 | FU-A | Fragmentation unit | |
| 29 | FU-B | Fragmentation unit | |
| 30-31 | reserved | | |
| | | - | |
| NAL Unit Type | Content of NAL Unit | NRI (binary) | |
| 1 | non-IDR coded slice | 10 | |
| 2 | Coded slice data partition A | 10 | |
| 3 | Coded slice data partition B | 01 | |
| 4 | Coded slice data partition C | 01 | |

A.1.2 MPEG-TS

A.1.2.1 Stream ID

| Stream ID | used for |
|----------------|--|
| 0xB9 | Program end (terminates a program stream) |
| 0xBA | Pack header |
| 0xBB | System Header |
| 0xBC | Program Stream Map |
| 0xBD | Private stream 1 |
| 0xBE | Padding stream |
| 0xBF | Private stream 2 |
| 0xC0 - 0xDF | MPEG-1 or MPEG-2 audio stream |
| 0xE0 - 0xEF | MPEG-1 or MPEG-2 video stream |
| 0xF0 | ECM Stream |
| 0xF1 | EMM Stream |
| 0xF2 | ITU-T Rec. H.222.0 ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream |
| 0xF3 | ISO/IEC_13522_stream |
| 0xF4 | ITU-T Rec. H.222.1 type A |
| 0xF5 | ITU-T Rec. H.222.1 type B |
| 0xF6 | ITU-T Rec. H.222.1 type C |
| 0xF7 | ITU-T Rec. H.222.1 type D |
| 0xF8 | ITU-T Rec. H.222.1 type E |
| 0xF9 | ancillary_stream |
| 0xFA - 0xFE | reserved |
| 0xFF | Program Stream Directory |